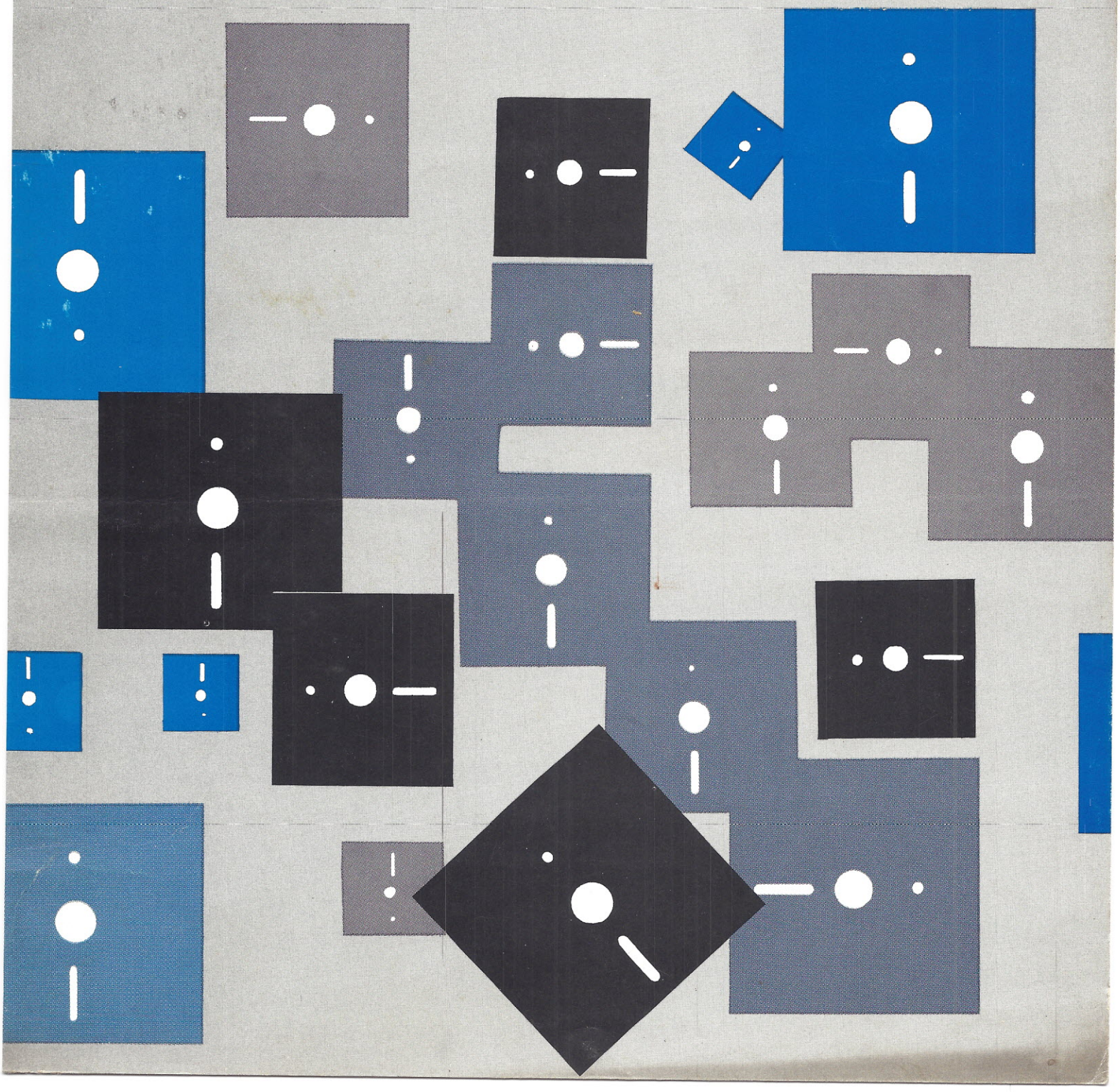
Lifelines™

The Software Magazine™

$3.00 November 1983 Volume IV, No. 6 (ISSN 02479-2575, USPS 597-830)



dBASE II™ made easy!

QUICKCODE™

The dBASE II Program Generator

Now dBASE II is made easy with Quickcode by Fox & Geller. QUICKCODE is a program generator, a computer program which writes computer programs.

FAST AND SIMPLE

With QUICKCODE you can generate a customer database in 5 minutes. Its that fast. All you have to do is draw your data entry form on the screen. It's that simple!

NO PROGRAMMING REQUIRED

QUICKCODE writes concise programs to set up and maintain any type of database. And the wide range of programs cover everything from printing mailing labels and form letters, to programs that let you select records based on your own requirements. There are even four new data types that are not available with dBASE II alone.

YOUR CONTROL

And since you work directly with your information at your own speed and your own style, you maintain complete control. Telling your computer what to do has never been so easy.

QUICKCODE, by Fox & Geller. Absolutely the most powerful program generator you've ever seen. Definitely the easiest to use.

Ask your dealer for more information on QUICKCODE and all the other exciting new products from Fox & Geller.

FOX&GELLER

Fox & Geller, Inc. Dept. LIF 001 604 Market Street Elmwood Park, N.J. 07407 (201) 794-8883

Lifelines

The Software Magazine

November 1983 Volume IV, Number 6

Lifelines (ISSN 0279-2575,USPS 597-830) is published monthly at a subscription price of $24 for twelve issues, when destined for the U.S. Canada, or Mexico, $50 when destined for any other country. Second-class postage at New York, New York; additional entry paid at Smithtown, New York. POSTMASTER, please send changes of address to Lifelines Publishing Corporation, 1651 Third Avenue, New York, NY 10028.

Editorial

by Edward H. Currie

In the Valley of the Blind the One-Eyed Man is King . . .

In a recent editorial, in a well known newspaper, the author suggested that the time had passed in which major successes in the application software marketplace could be the result of the efforts of the entrepreneur. The example given to support this claim was that of Lotus 123. After all, this product had risen to preeminence in record time and thus set standards with which the individual software developer could not hope to compete. The basic argument presented was that this was now the domain of the high rollers.

Sales figures were given for a number of software manufacturers in an effort to further substantiate the claim that unless you were backed by lots of money there was no point in entering the race.

And to some extent this is perhaps the case. At the moment the vast majority of potential customers for software is relatively naive about microcomputers and micrcocomputer software and as a result they are easily swayed by expensive ads, fancy brochures, fancy packaging and the rubber stamp endorsement of the publication media.

Furthermore, dealers have neither the time nor the interest to learn the intimate details of even the most trivial of the vast numbers of applications packages. Therefore, they too quickly line up behind one product or another depending upon the amount of advertising and PR which they encounter.

Now there is no question that Lotus 123 is an extremely fine package or that Lotus has done much to move the industry forward. What is at issue here is whether or not this is a bellwether for the smaller players in this game.

The fact is that what Lotus is offering the world is being backed by a powerful marketing operation supported by lots of bucks. If we were in the automobile business in which performance is largely of academic interest then perhaps the Lotus approach could be extrapolated to the rest of the industry. But clearly we are not.

The installed base of microcomputer users is growing at an ever increasing rate, as is the diversity of demands for applications programs and feature, function and benefits. As the now unwary enduser moves out on the learning curve he will undoubtedly begin to demand better and better products with little concern for such diversions as fancy packaging, expensive ad campaigns, etc.

The point is as simple as this . . . if you are a software author, ignore the roar and din of the market and press on with the development of your products. What we are observing is a temporary diversion which will give way shortly to a more rational and practical buying criteria on the part of the end user. It works for Lotus because they are offering a fine product. It won't work for many of the others, however. It's up to you to continue the high quality and innovative programs that the software publishers can market effectively for you.

The telecommunications market continues to develop deep inroads into the microcomputer marketplace due to availability of relatively inexpensive 1200 baud modems and telecommunications applications packages. There is, however, little to indicate that a standard protocol for file transfer will emerge in the near future. Ward Christensen's MODEM7 protocol continues as the *de facto* standard. It is likely that a variety of protocols will continue to be supported in the interim.

Unfortunately, microcomputer telecommunications evolution is being slowed significantly by the failure of the various information and database services to make their services and benefits known to the microcomputer users. This may well be a phase lag problem.

The microcomputer world was viewed by many as interesting but not really worthwhile from the standpoint of those selling services. With IBM's entrance the market was suddenly viewed as "legitimate." This meant among other things that the market was now substantive enough to be taken seriously. Thus, various traditional maintenance organizations began to offer maintenance contracts for microcomputers, Fortune 1000 companies began acquiring micros in large numbers, etc.

This should mean that soon there will be a large number of databases accessible to micro users, both professional as well as those looking for entertainment. Imagine being able to access a telephone directory, for example, that had every phone number in the United States with associated names and addresses.

It is interesting to note that Teletext has suffered a significant loss of momentum due, in part, one would suspect, to the fact that most of us are interested in an interactive environment as opposed to one which just supplies games and spews forth random information about football scores, movies at local theaters and race results.

Couple the existing technology with laser disks and we can begin to see that there may be some exciting developments on the horizon.

The response to the recent editorial in this column in which mention was made of the two meter band was, to say the least . . . interesting.

Previous editorials have stressed our concern for the future of ham radio. While some of you seem to be convinced that this avocation is growing there is a considerable body of evidence to indicate that it is in fact contracting. The fact that absolute numbers may grow does not mean that one can ignore the percentages. And percentagewise it is contracting at an alarming rate. One has only to visit the local magazine store to discover

(continued on page 28)

Feature

Some Spreadsheets

by Robert P. VanNatta

Spreadsheets, calc-clones, or whatever you want to call them, have proliferated all over the place. The following ruminations are intended to provide some insight into three of them. The idea is not so much to explain how to use a spreadsheet as it is to provide some information that might be useful if you are attempting to select a spreadsheet for use.

The particular versions and products that are the subject of review here are:

1: **CalcStar version 1.2.** There is a later version (1.4) out but it was not actually available at the time this was written. This product is marketed by MicroPro International, the makers of WordStar. The new version is supposed to have increased capacity and screen scrolling.

2: **SuperCalc version 1.12.** (This is thought to be a current version). SuperCalc comes from Sorcim and is widely marketed.

3: **Perfect Calc version 1.0.** (A later version is in the works but only a beta test version was available to this writer. The difference is that some bugs are repaired in the later versions). Perfect Calc is from Perfect Software. It is most famous for being bundled with the Kaypro II, but is available elsewhere as well.

What's a spreadsheet?

In case there is still someone who doesn't know what a spreadsheet is, I shall attempt to explain what one is and what it does. Fundamentally, a spreadsheet is an electronic version of the familiar large yellow columnar pad which is available from your local office supply house. Like the big yellow pad, it is arranged in rows and columns. The columns can be any width you desire. Each point on the sheet defined by the intersection of a row and a column is called a 'cell.' By tradition, the columns across the top are labeled alphabetically; a, b, c, etc. The rows are numbered down the left side; 1,2,3, etc. The cell in the upper left-hand corner would be called 'a1' and the cell below it would be called 'a2'. The cell to the right of 'a1' would be 'b1'.

Any cell may generally contain one of three things. First, you can use any cell for a label. As the name implies, this is simply a word, or if you have made the column wide enough perhaps several words whose function is to describe or identify something. The label on the electronic spreadsheet is no different than the column heading or row label that you might use on your big yellow pad, and its function is the same.

The second thing that might be in a cell is a number. Just as you may arrange rows or columns of numbers in your big yellow pad, you can enter rows or columns of numbers on the electronic version of your yellow pad.

It is the third alternative for a cell that really sets the electronic spreadsheet apart from the yellow pad. The third thing that may be placed in a cell is a formula or equation. It sounds complicated, but actually the concept is quite straightforward. Suppose, for example, that you want to use the third row to display the totals of the first two rows. The exact syntax for entering formulas varies some from program to program, but something like 'a1+a2' entered into a cell will cause the cell containing the formula to display the contents of cell 'a1' plus the contents of cell 'a2'.

The true beauty of the electronic spreadsheet begins to seep through when you figure out that you can go back and change the number in cell 'a1' or cell 'a2' and have the computer recalculate the totals.

The typical spreadsheet is flexible enough to be useful for almost any application which involves the manipulation of numbers. Tasks that were formerly done by hand on big yellow pads are obvious applications, but only the ingenuity of the user really limits the use of a spreadsheet. Typical applications include budgets, payment schedules, balance sheets, profit and loss statements, statistical analysis, bid figuring, tax returns, and stock market analysis. In fact, the spreadsheet is quite suitable for any task which might otherwise be done on an adding machine or calculator.

Selecting a computer

If you are going to use a spreadsheet, you need to think about having a computer to run it on. I mention this because a computer configured for a spreadsheet is quite a different computer than one configured for accounting or wordprocessing. The action with a spreadsheet program takes place in memory. This means that if you want a computer suited to a spreadsheet, you should spend your money buying extra memory boards for it, not extra disk drives. This strategy is almost the exact opposite of the wisdom which must be applied for general accounting activities. It is sort of the norm for accounting packages to run in as little as 48K of memory and at the same time need two to five megabytes of disk storage for even a modest sized business. With an accounting package, once you get the critical mass of memory sufficient to get it to run at all, additional memory will provide no additional performance, but large fast disk storage devices will often do wonders.

By contrast, the capacity and usefulness of a spreadsheet is directly tied to the amount of available memory. As we shall see later in this article, some spreadsheets make more efficient use of memory than others; but the general message is: More memory is better! Minimal disk storage of a few hundred kilobytes will likely prove adequate. These comments, however, do not apply to Perfect Calc, as we shall see later.

The test equipment

The work that I did here was with a

Radio Shack Model 16. Operating systems tested included Lifeboat's own SB-80, and CP/M 2.2 with the Lifeboat bios version 2.25d, CP/M 2.2e from Pickles and Trout, and CP/M 2.23 from ATON. The latter is the least well known, but the most interesting in that it supports bank switching. The ATON CP/M is sold by ATON International of 260 Brooklyn Ave., San Jose, CA 95128. ATON testing was done with the bank switched version set up with 128K of memory. (They also have a non-switched version.) This gives 62K of TPA and full track buffering. Unfortunately, the long promised CP/M plus was not available as I wrote this, but the ATON switched version should approach CP/M Plus performance.

Installation

Maybe it is too obvious to mention, but a prime consideration in selection of a spreadsheet package needs to be related to how well it works on your particular computer. A spreadsheet necessarily needs to make use of most of the extended terminal features. A good installation will make use of both the arrow keys (for moving the bug from cell to cell), and inverse video or highlighting to identify the bug (e.g., the cell ready for input).

As I can attest from the variety of operating systems that I use, there is a great lack of uniformity in installation quality from system to system. For example, SuperCalc installs very nicely on Pickles and Trout CP/M, complete with highlighting and functional arrow keys; but on Lifeboat CP/M I was not able to get either the highlighting or arrow keys to work, making for a much less desirable package. Conversely, Perfect Calc could be installed on Lifeboat CP/M with arrow keys and highlighting but I couldn't get it to work at all under Pickles and Trout CP/M, even though it was listed on the installation menu. The bag was mixed with ATON, and Perfect Calc. It wasn't listed on the installation menu, but could be made to work if you disabled the highlighting features.

CalcStar provides custom definable cursor control keys and, in fact, has a rather good custom installation routine. However, the menu listed installation routines do not attempt to enable the arrow keys. The disappointment is that multiple definitions of commands are not allowed. Thus, if you define your arrow keys, you must forfeit use of the traditional Wordstar 'Star' pattern of cursor control. The only highlighting is on the '<>' which is used to mark the current cell. Each cell has a reserved space at each end of it for the markers so the absence of highlighting with CalcStar is not as serious as it is with Perfect Calc and SuperCalc. The latter two do not have a reserved place for the cell markers (none is needed if the whole cell is highlighted) and where highlighting is not installed the cell marker poaches on adjoining columns. It is much less pleasant to work with than the highlighted versions.

A summary of my installation comments in table form is as follows:

PERFECT CALC:

Best installation routine. Allows multiple definitions of commands. Highlighting much needed, but not handled by menu installation. Custom installation easy to use. Highlighting doesn't work right on ATON and Pickles and Trout implementations of CP/M 2.2. Uses a disk workfile to permit 3000 entries for all operating systems. Performance somewhat dependent on disk speed. On a floppy system, it bogs down in file writing long before it runs out of space. The 128K track-buffered bank-switched ATON CP/M reduced recalculation time of a 3000 entry sheet from 2.5 minutes to about one minute. The only good installation was with Lifeboat CP/M.

SUPERCALC:

Good menu installation routines, but the custom installation routines could be improved. I have not been able to get arrow keys installed on a custom basis. Pickles and Trout menu installation includes both arrow keys and highlighting. With ATON, I got everything to work by redefining the operating system to emulate Pickles and Trout, but never did get arrows to work with either Lifeboat CP/M or SB-80. It seems that many installation menus that include Lifeboat CP/M as a specific item, patch it in according to the Lear-Siegler ADM3a routine and omit highlighting. Writer's choice is for Pickles and Trout CP/M. Capacity about 1200 entries on standard CP/M, up to 1900 entries using ATON Bank-switched CP/M.

CALCSTAR:

Installation menu ignores cursor keys, but above average custom installation routine allows key definitions (at the expense of disabling standard keys). Use ADM31 option for Lifeboat CP/M. Lack of keyboard buffer in SB-80 shows up. Groggy on Pickles and Trout. Bank-switched ATON gives capacity of 635 entries compared to about 470 on standard CP/M.

Using the spreadsheets

Once you get over the installation problem (and I would judge it a serious headache on both Perfect Calc and SuperCalc), the next problem is trying to learn how to use the chosen spreadsheet. Each of these programs has a help facility for the novice. Each works in quite a different fashion, however.

Calcstar presents a list of commands on the top part of the screen *à la* WordStar and a one-page summary of the formula operators available. If you already know how to work Calcstar, this is enough to be a handy reminder, but as a beginner you will find it completely inadequate. With Perfect Calc, you can trigger a help menu on the bottom of the screen with a question mark, and from there can select the topic of interest. The information is quite detailed, but presupposes that the user has sense enough to know what he needs the help for.

Clearly, the best help facility is provided with SuperCalc. The entry of a question mark at almost any point in the program will display a help screen relevant to the task at hand. This 'Smart help,' although less explicit than that of Perfect Calc, is, in this writer's opinion, a better deal. If, for example, you are trying to copy a cell to another location and can't figure out how to make the copy facility work, a push of the question mark will give you help on the copy facility, not a master menu listing umpteen help choices.

Beyond the help facility, there are some differences in the way information is input to the spreadsheet. Each of the spreadsheets accepts information which can be roughly categorized about four ways. First there are labels. A basic programmer would recognize those as strings. They are really just messages that you can

type onto the spreadsheet so you can keep track of what you are doing. Column headings would be a typical example.

With both CalcStar and Perfect Calc, anything you type that begins with an alpha character (as distinguished from a numeric) will be treated as a label. Supercalc expects labels to begin with a double quote, and, if the quote is missing, treats the value as a formula error. This writer finds this characteristic of Supercalc irritating.

The second common input is a numeric literal such as '123'. These are handled uniformly and all programs allow you to just type them in.

The third major type of input is the equation or formula. Each program has its own way of identifying these. An equation is a message designed to inspire the mathematical routines of the program to make some calculation. A typical equation would be an instruction to add two cells together such as 'a1+b1'. As mentioned above, anything typed in that starts with an alpha character is assumed by SuperCalc to be an equation. Thus 'a1+b1' typed in (without the quote marks) would work with SuperCalc, but would be taken as a label by the others. Perfect Calc expects all equations to begin with an equal sign ('='). Thus, in Perfect Calc, you would type '=a1+b1'. Calc Star has an even stranger method. In CalcStar, an equation is identified if it starts with a numeric character and contains alpha characters in its text. Thus, with CalcStar, you must enter '+a1+b1' to get the answer. My inclination is that the Perfect Calc approach is the choice of the lot.

Hand in hand with the equations are the formula operators. All of the spreadsheets will add, subtract, multiply, divide, sum columns, count the number of entries in a column, extract averages and perform numerous other mathematical gymnastics. Likewise, each will perform certain logical operations such as the IF-THEN-ELSE statement. The syntax for the functions varies from program to program; however, the syntax used by SuperCalc and Perfect Calc is very similar, and reasonably understandable. Some of the syntax in CalcStar is downright obscure. For example, '+sum(a1>a22)' uses the '>' to mean 'to'; thus the command will add up all the numbers in column 'a' from one to 22. In other contexts, however, the '>' is the traditional relational operator.

SuperCalc has the longest list of built-in formula operators, and Perfect Calc is a close second, lacking only 'Pi', and some of the more obscure trigonometric functions.

The fourth type of input common to spreadsheets involves commands. These include the housekeeping tasks, such as moving the cursor, scrolling the screen, loading files, printing files, formatting entries, and moving data from area to area. It is in this area that these programs vary most widely. Both SuperCalc and CalcStar have a command mode. Like WordMaster, Spellbinder or Vedit, when you enter the magic key (a ';' in the case of CalcStar and a '/' in the case of SuperCalc) the program shifts gears and expects some command. A 'Q' from the command mode will exit the program, for example. With both SuperCalc and CalcStar the default method of moving the cursor around the screen is the familiar WordStar pattern of ↑E for up, ↑X for down, etc. Entry to the command mode in these programs will expect you to select one of about 20 commands, and upon completion return you to the video mode.

By contrast, Perfect Calc handles the commands with the use of two- and three-key control sequences and escape sequences which are logically not unlike WordStar. There is no command mode as such, but there are 60 or more commands. Many wags have said that WordStar was difficult to learn because of its many control sequences which seem to be randomly assigned to keys. If this is a valid criticism of WordStar, it is equally applicable to Perfect Calc. The main difference is that the Perfect Calc people used a different random number generator. Any similarity in the Perfect Calc and WordStar Control sequences is only imaginary. Perfect Calc features both horizontal and vertical page scrolling. This is a wonderful feature and I use it a lot, but unfortunately the command for scrolling down is a ↑V and the ↑C beside it is for (c)utting out a line. If you are a seasoned WordStar user, as I am, you will perhaps understand why my spreadsheets have missing lines here and there, and if you aren't, well, forget it!

Like SuperCalc, Perfect Calc supports multiple windows, but it goes one step farther and allows multiple files as well. You are only allowed to have two windows, but may have several files which may be assigned at will to either window. Copying and replication from one file to another is possible. You may also pass information from one file to another. For example, if you have set up an income tax return template, various schedules might be assigned different files, and the results of a supporting schedule could be passed to the main schedule.

Reportedly, the new version of CalcStar (version 1.4) that was released about June of 1983, but which was not available to this writer, has a scrolling feature, and a cell capacity that is about doubled from its current meager amount. However, it is this writer's notion that when it comes to commands, Perfect Calc has the other two beat all to pieces. Essentially, I would characterize the commands as maddening to learn, but fast and powerful once mastered.

The faults

Every program has its weaknesses as well as its strengths. It is the reviewer's toughest job to identify those weaknesses, as they are often in the form of 'non-features' rather than outright bugs. The biggest single drawback to CalcStar is related to its extremely limited capacity. The new version (1.4), which is not tested, claims doubled capacity. Even if this is true, it is still considerably less than SuperCalc. It is the least expensive of the three, and is plain vanilla. It is my notion that SuperCalc and Perfect Calc are very competitive. I have had frustrations installing both of them. Both will function on terminals with and without highlighting, but if you have a terminal that supports highlighting, I would lean toward the one that will install with that feature working. Perfect Calc is the newest of the programs and still shows a rough edge here and there. It is written in 'C' so it may be showing up on other systems soon. It has the greatest cell capacity, but there is a price for that capacity. It uses virtual memory management to attain the great capacity. This is a mouthful. What really happens is that Perfect Calc writes portions of the spread-

sheet out to disk when the file size gets to be too much for memory. A 3000 entry spreadsheet (or combination of several spread sheets if multiple files are loaded at the same time) occupies a 64K workfile. If you go wandering around the sheet, guess what must be constantly rewritten, and guess what you are doing while this process is going on?

Perfect Calc is a whale of a program, and ought to be matched with a whale of a computer. When you recalculate a 3000 entry spreadsheet, the squirrels really get busy inside your computer. As noted in the table earlier in this article, the addition of track-buffering and bank-switching are great benefactors to Perfect Calc. They help, but if you really want to make practical use of the extra capacity, you are going to need to have a state-of-the-art 8-bit computer. By this, I mean a memory disk and a Z80B or 8085 processor clicking at some exotic speed.

Perfect Calc has for the last year or so been bundled with the Kaypro II portable Computer. It is my notion that this has been a sad mismatch. Low density 5¼" drives are not my idea of high performance, and the need to shuffle the workfile quickly bogs the poor little Kaypro down, making both look bad.

SuperCalc works in memory only and is not drive-dependent. I would, therefore, say that it might be a better choice for systems with low performance drives.

All I can really say about CalcStar is that it works and is less expensive than the others.

Conclusions

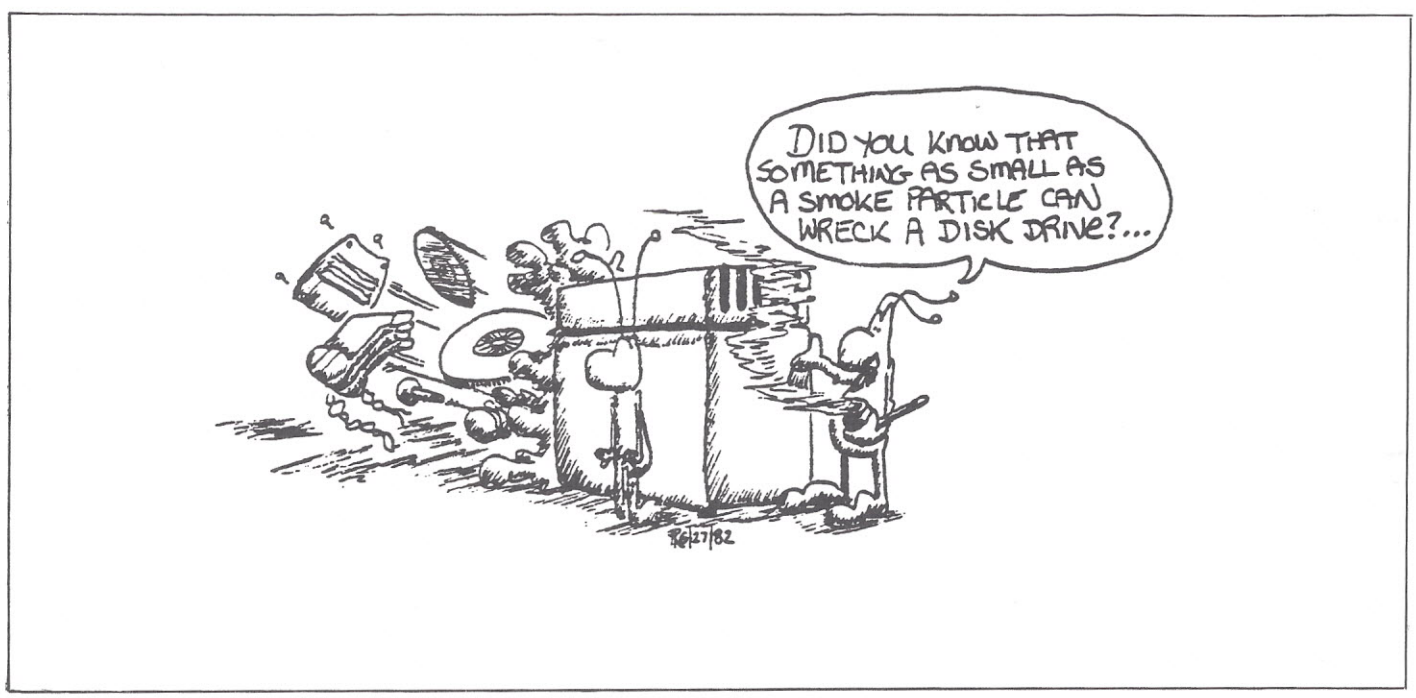
The Calc Clones are not all the same. At least based on my observations, performance is likely to vary widely from computer to computer and even from operating system to operating system. My advice is to select a spreadsheet that actually works on your computer system. You don't have my system or my needs. If you pick it because I like it (or don't like it), you might be disappointed. For example, some months ago, (July, 1982) the much respected *Lifelines* contributor Ward Christensen penned a review for a little program editor called VEDIT from Compuview Products of Ann Arbor, Michigan. One of his biggest gripes about the product, as I recall, was that it seemed slow. VEDIT came to me not long after that for evaluation for another magazine (I also write regularly for a magazine called *TWO/SIXTEEN*, Box 1216, Lancaster, PA 17603, which is dedicated to support the Radio Shack business computers). Having read Ward's remarks, I opened my VEDIT box with the utmost of care. After all, I didn't want the pooch to bite me.

When I got VEDIT running, I was amazed. It seems that on my equipment, it supports a memory mapped installation and is the fastest editor that I have. The command keys as issued seemed a bit strange to me, but they were all user definable; so I redesigned the keyboard commands to match (almost) those of WordStar, and I have myself an editor which, at least on small files (e.g. those that will fit in memory) will run circles around either WordMaster or WordStar.

These differences in opinion cannot be accounted for by suggesting that one of us can't tell fast from slow. Rather, the difference is in the hardware on which the evaluation is done.[1] A program that looks good on one system can be a real mutt on another. I don't have a room full of different computers to try these programs out on, but I do have a desk drawer full of different operating systems. I never cease to be amazed at the widely differing results that even a swap of operating systems can bring.

FOOTNOTE

1. The other reason is that I have found a niche for it not considered by Ward Christensen. He tested it as a primary program editor. I use it as a secondary editor for small jobs particularly when I don't want to load up WordStar. It is ideal for editing short messages for telecommunications transmission, and editing short programs where quick "ins and outs" are the primary concern. It will not set high-order bits and can be configured to expand tabs into an appropriate number of spaces. These features seem to keep some of the systems that I communicate with from having the hiccups.



Feature

Learning The C Language With C-System's C Compiler

by Ron Watson

While learning a computer language is easier for most people than learning a "natural" language, many of the same skills are necessary. One cannot consider himself proficient in either until he has reached a level of familiarity that allows him to "think" in the language. A good compiler will assist the learning process by acting as interpreter and as a kind of reflexive tutor: it can tell you what you did wrong, but will not be able to offer much advice on how to correct the mistake.

I chose to learn the C language because I needed a tool to develop applications for the IBM-PC. Assembly language is fun, but too time consuming; BASIC is easy to use, but has inadequate program control and data structure tools; FORTRAN and COBOL seem too clumsy for a micro. That leaves Pascal and C. I chose C because it is native to the Unix operating system, which seems to be the direction in which micros are moving, and because of the simplicity of its specifications. This last is exemplified by the Kernighan/Ritchie manual: 225 pages including tutorial, reference and index. Compare this with the eight or nine hundred pages of documentation needed for some languages.

The c-systems compiler seems to be a good choice to learn the language because it adheres very rigorously to the K&R standards, at $200 is relatively inexpensive, and for another $200 you can get an interactive debugging tool. An interactive debugger allows you to supervise the running of a program, stopping and starting it at will, displaying results, and altering variables without making source level changes to the program. A good debugger can reduce program development time by an order of magnitude. Since one usually makes more mistakes when first learning a language, this tool is an excellent investment for learning, and no other C compiler offers one.

To get started, I first read quickly through the K&R manual to get a general idea of the language. Then I went back to the beginning, armed with my new compiler, re-reading for detail and doing each of the exercises in the tutorial.

The compiler comes with two diskettes and two manuals, one each for the compiler and the debugger. The compiler documentation is not particularly good; it's disorganized and has no index, so I had to read through it several times before I learned where to find things. I started with version 1.15 of the compiler, which incorporates a new precompiler. The body of the documentation was not updated to reflect this change; instead a three page errata sheet was inserted. Installing the system turned out to be more difficult than necessary because of the general sloppiness of the documentation. The compiler parts are not copy protected, so installation should be a simple matter of copying the appropriate files to a work diskette. The difficulty lies in identifying the appropriate files. There is not enough room on a nine-sector, two-sided diskette to hold all the files included on the distribution diskette along with copies of the IBM assembler and the operating system files, so knowing which files are needed is important. I finally wound up with the compiler, pre-compiler, object library, linker and assembler on one diskette, which I use in drive A, and the debugger library and my own programs on drive B. This allows a complete compile and link without changing diskettes. There was not enough room left on drive A for my text editor, so I put it into a RAM disk. This method worked well as long as I took care to delete files from drive B when they were no longer needed.

I should mention that the c-systems compiler does not generate machine code directly. Instead it outputs an assembly source program that must in turn be processed by the IBM assembler to produce an object file. This considerably increases the elapsed time necessary to compile and test a program. The small, one and two page programs in the K&R tutorial needed two to three minutes to go through the whole process on a two-diskette, 256K system; this increased to four or five minutes if the debugger code was being used. I found this to be an acceptable delay, but others may not.

Once through the minor trauma of installation, I was ready to get to work on the K&R tutorial. Being of a logical inclination, I started at the beginning with the first example program in the manual. This turned out to be a bad idea. The first example program contains about 15 lines of code that convert temperatures from Fahrenheit to Celsius. Sounds simple enough, except that K&R quite reasonably use floating point numbers to make the calculations and this compiler does not support them. After I discovered this, I began to investigate what other C language elements might be missing. Besides floating point, the manual mentions casts and extended pointers. (A cast is a method to tell the compiler that a variable which has been declared as one type is to be used in a particular statement as another. It is slightly equivalent in use to the FORTRAN "Equivalence" declarative.) I have since received a copy of version 1.16b of the compiler, and while it arrived with no additional documentation, it appears to include support for casts, which leaves only floating point and extended pointers missing. The publishers assured me that floating point support will be available later this year. Thus comforted that I would not encounter any serious compiler limitations, I proceeded through the tutorial.

The first chapter went quite smoothly with the compiler processing each example and problem as expected from the K&R text. The compiler includes a facility for input and output redirection which allows the pro- ►

grammer to run a progam once with input from the keyboard and again with input from a disk file. File redirection is done with the same syntax used in DOS 2.0 and Unix. This feature is a great aid to testing and debugging.

In Chapter Two, there is a problem requiring a program to convert a string of hexadecimal characters to an integer value. I had trouble getting it to work properly, so I used the interactive debugger (c-systems calls it "c-window") to find the problem. You use the debugger by specifying a special parameter when the program is compiled and including an additional object library in the link step. This adds a minute or two to the preparation time for a small program, and because c-window is resident with the subject program, it adds about 50K to the EXE program size. It also increases execution time, but that is of little consequence during testing. When the subject program is run, the first prompt that shows up is a ">>" request for commands from c-window. At this point you can enter commands that request break points to be set to stop program execution later, display variable contents, or evaluate C expressions. All commands may refer to the program by source line number and to variables by their declared name. Automatic commands can be entered that are executed when certain conditions are met. They are expressed in C syntax using program names already familiar to the programmer. The "go" command causes execution to commence, or the "step" command can be used to execute the program one source line at a time. This tool ranks with the best I have seen on minicomputers and mainframes for power and ease of use. Needless to say, I found my error rapidly, and without having to add any ugly little print statements to my source code.

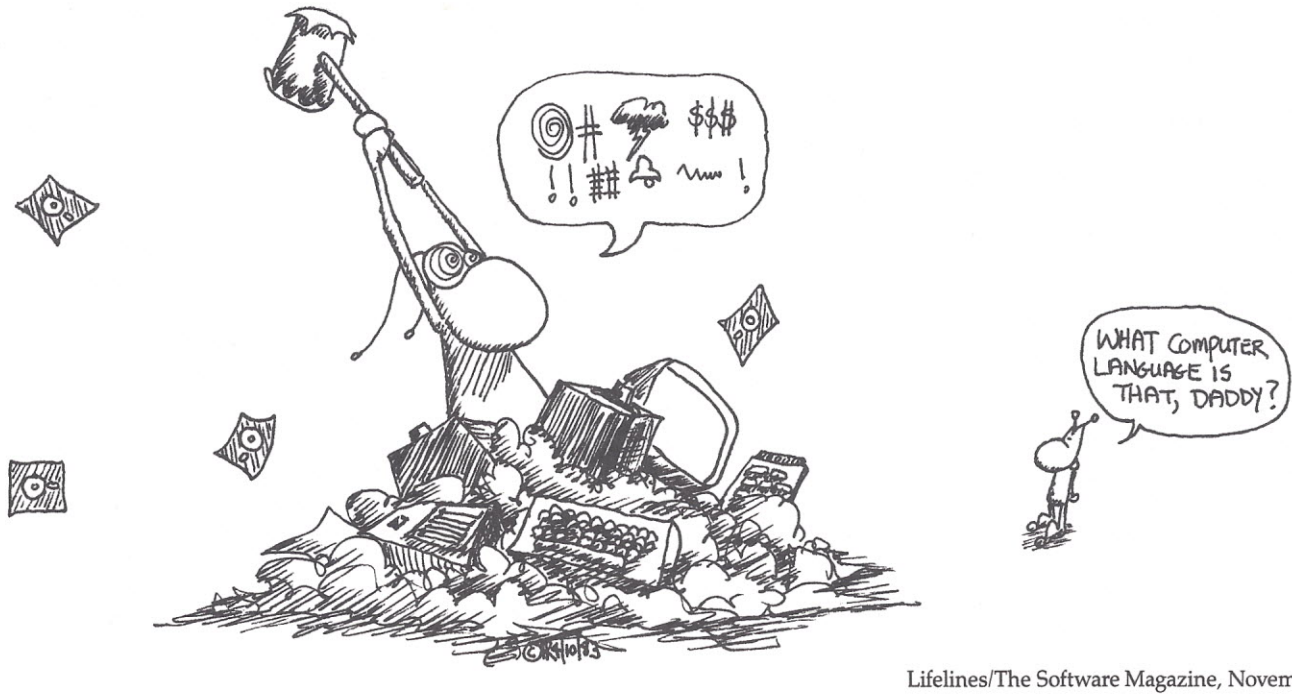
As I progressed farther into the tutorial, the problems got more complicated and the programs got correspondingly longer, and I made mistakes more often. How a compiler responds to a syntax error can have a profound effect on the time needed to write a program. This compiler responds accurately to an error, but has a tendency to get confused and display several additional messages for each error in the program. The messages are understandable and correctly indicate the line in error. If a more precise location of the mistake is needed, you can edit or print the compiler-generated assembler program, which will contain comments that indicate the exact column containing the error. This is not often necessary, but can be a great help to the student programmer.

One element of a C compiler that distinguishes it from others is the content of the function libraries provided. C does not contain any syntax for input/output, as these facilities are meant to be provided by functions that are included in source or object form with the compiler. This compiler makes these functions available in an object library that is linked with the output from the assembly to produce the final program.

Most of the functions mentioned in K&R are included in the library, as well as additional functions to allow the programmer access to DOS function calls and the I/O ports in the PC. This means that nearly anything that can be done in assembly language can be done in C with this compiler.

The distribution diskette contains two versions of this library, called the "small" and "medium" versions. The small one is used when the object program size is less than 64K; the medium one, of course, is for use with larger progams. The principal difference between the two is the type of machine language "call" instruction used to transfer control between subroutines. While this feature does make programs possible with large code segments, no provision is made to address more than 64K of data. If I had my choice, I would rather be able to have large data areas and small programs, but this is not the case. This problem seems endemic with compilers for the PC. The difficulty with C may be a shortcoming in the language itself.

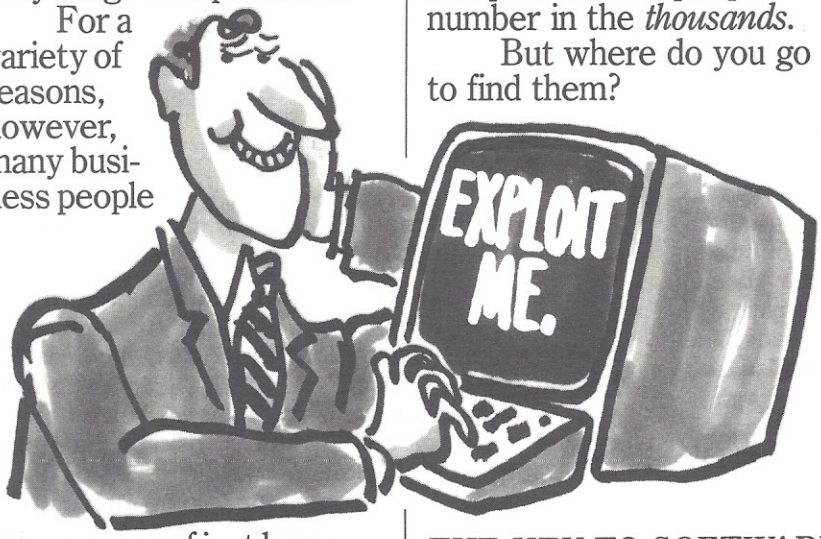
In summary, I found this to be an excellent product that delivered everything that it promised. I especially would recommend it and its companion c-window debugger to anyone wanting to learn the C language, and I would recommend C to anyone planning to develop programs on the PC. ■



YOU SPENT $4,000 ON A PERSONAL COMPUTER. FOR ANOTHER $12.50, YOU CAN GET YOUR MONEY'S WORTH.

Today's personal computers have an extraordinary range of capabilities.

For a variety of reasons, however, many business people



are unaware of just how much their computers are capable of.

As a result, they aren't realizing the full potential of their investment.

THE KEY TO GREATER PRODUCTIVITY IN A WORD: SOFTWARE.

Computers do the work. Software does the thinking.

Expanding the amount of work a personal computer can do is merely a matter, then, of gaining access to a broader array of software.

And the software programs available to business and professional people number in the *thousands.*

But where do you go to find them?

THE KEY TO SOFTWARE IN A WORD: *LIST.*

LIST is the first publication that puts software first.

It contains articles by some of the most respected names in the computer field. Written to help you get the most out of your personal computer. No matter what brand it is. No matter what you need it to do.

More importantly, *LIST* contains the *LIST Software Locator,*™ a comprehensive guide to over 3,000 personal computer programs—conveniently indexed by application, industry, operating system and hardware. You'll find detailed descriptions of applications software that pertains specifically to the type of business you're in. And the type of needs you have.

LIST is sold at leading computer stores and bookstores. Or, you can phone our toll-free number (1-800-821-7700, Ext. 1110) or send in the coupon below, and receive a copy by mail. The price, exclusive of postage and handling, is $12.50.

Which, when you think about it, is a pretty small price to pay for something that can maximize a much larger investment.

LIST *is published by Redgate Publishing Company, an affiliate of E.F. Hutton.*

I'D LIKE TO GET THE MOST OUT OF MY PERSONAL COMPUTER.

Please send me _____ copies of *LIST* at $12.50 a copy plus $2.00 each for postage and handling. (Tax will be added where applicable.)

☐ VISA ☐ MasterCard (Interbank No. _____)

Card No. _____ Exp. Date _____

Signature _____

Print Name _____

Address _____

City _____ State _____ Zip _____

Send to *LIST,* Redgate Publishing Co., 3407 Ocean Drive, Vero Beach, FL 32960.

Or phone, toll-free: **1 800 821-7700 Ext. 1110**

MasterCard VISA

LIST™

The Software Resource Book For Personal Computer Users

Feature

PL/I From The Top Down

By Bruce H. Hunter

This is the fifth of a continuing series of articles about PL/I Subset G rewritten from my book entitled PL/I From The Top Down. *The subject of this article is,"Edited I/O, Plain And Fancy." Because my book is a casual approach to PL/I and thus is not a comprehensive coverage of the subject, in each installment I like to make reference to other books on the full set for those interested in further study.* Structured Programming in PL/I and PL/C, *by Peter Abel (Reston, 1981), is a good book, and it has some sections on edited I/O that are well written which will supplement this article nicely. The PL/I compiler used for this article series is Digital Research's PL/I-80.*

CHAPTER FOUR — EDITED I/O, PLAIN AND FANCY

PL/I is superbly equipped to produce edited input and output. Whereas list I/O has inherent limitations as to the possible formats available, edited I/O gives you the ability to precisely format input and output to your own needs. The origins of edited I/O go back to the "bad old days" of card I/O and FORTRAN where rigid formats were a programmatic necessity. Although a number of anachronistic carry-overs are present for card input in the full set of PL/I, a few of which are also present in PL/I Subset G, for certain applications the virtues of edited I/O far outweigh its vice of minor inconvenience. Other language implementations for micros simply don't offer you the choices available in PL/I. If you prefer not to take advantage of the features of edited I/O, you always have the option of falling back on list I/O. When you understand the programmatic possibilities edited I/O lays at your fingertips, however, you will start to appreciate the elegance of PL/I as a programming language, even in the micro implementation.

Let's take a look at what some of the other languages have available as edited I/O. If we consider the principal programming languages, Pascal, C, COBOL, FORTRAN, BASIC, and PL/I, we can eliminate Pascal right off the bat because it has extremely limited formatted I/O capabilities. C is a horse of a different color because C has a unique and very powerful set of formatted I/O functions. It is the only language that allows right and left justification of both character and numeric data with ease. However, because it does not have anything resembling PL/I's picture format and because it does not have an actual column function, it still does not measure up to PL/I. Although COBOL has excellent edited I/O capabilities, programming in COBOL is like waiting in a horrendously long line to get into a restaurant. No matter how good the food is, it takes too long to get there. Personally I don't even consider it a "contender" as a productivity language. FORTRAN is a formidable "number cruncher," and it's ideal for specific scientific and mathematical purposes, but it's not suitable for other programming purposes because of its inherent limitations in handling strings and its lack of structure. That leaves us with BASIC and PL/I. The closest thing that BASIC has to edited output is the *print using* command which allows for a certain degree of output control. You can specify width and location of character fields, so columns are readily obtainable. BASIC allows you to have a dollar sign at the beginning of the number as well as a plus or minus sign at the beginning or end of a number:

```
$ 1234.56+
$ 1098.57-
$+0971.01
$-0078.32
```

There are limitations to BASIC's *print using* command, but all things considered it does quite a respectable job.

Some of the edited I/O features of PL/I include the picture format which corresponds to BASIC's *print using* command (and is close to COBOL's *picture*). Whereas BASIC does a good job with edited I/O, PL/I's picture format can walk on water. Edited I/O in PL/I gives you left justification of character fields, right justification of numeric fields, zero suppression, repeat factors, virtually complete spacing control, imbedded control characters including formfeeds and linefeeds and that's just the beginning. If you were doing an accounting program and you wanted to signify "cr" for "credit" and "db" for "debit" rather than "+"or "–" at the beginning or end of the output fields, there's no problem! Let's repeat the above BASIC example as it could be output in PL/I using its zero suppression feature and printing "cr" and "db."

```
$1234.56db
$1098.57cr
db$ 971.01
cr$  78.32
```

In addition, PL/I can control input data formats as well as output data formats to various specifications including scale and precision. You even have control over the type of data representation you wish to specify — fixed, bit, exponential, or alpha. Thus it is possible to do all of your calculations in floating point and have them printed out in scientific notation. Like FORTRAN, PL/I can even specify remote format. But rather than tantalizing you further with generalities, let's get down to specifics! Before going right into a discussion of edited I/O, I want to quickly review edit statements. Next, we'll look at the significance of scale and precision in edited I/O. Lastly, we will briefly examine some of the types of formats and what they do.

Quick review of edit statements

Remember the edit statement? We talked about edited I/O (also called formatted I/O) in the second installment of this series (July 1983 issue). Here are some of the basic output forms:

1. put file (file_id) skip (n) edit (expressions) (formats);
2. put skip (n) edit (expressions) (formats);
3. put edit (expressions) (formats);

And here are some of the basic input forms:

4. get file (file_id) edit (expressions) (formats);
5. get skip (n) edit (expressions) (formats);
6. get edit (expressions) (formats);

Referring to the edited I/O program statements above, for *expressions* you use constants, variables, or expressions using both. The keyword *file* and the file id *file_id* (examples 1 and 4) can be omitted if the default files, *syslist* and *sysprint*, are preferred (examples 2, 3, 5 & 6). When *skip (n)* is used on output (examples 1 and 2), this signifies that *n* number of lines will be skipped. If *skip* is used alone, one line is skipped. (When using Digital Research's PL/I-80 and PL/I-86, *skip* specifies that a linefeed be performed to which the CP/M operating system adds a carriage return, thus the CR/LF pair.) When *skip* or *skip (n)* is used on input (example 5) *n* specifies the total number of "carriage returns" desired before putting the entirety of the data input into the storage locations (i.e. variables, constants, addresses) specified in the declarations.

The point I want to drive home is that *edit* is the keyword telling PL/I to look at *formats* in the edit statement for format codes, control formats, picture formats and so forth. In the following edit statement

```
get edit (name) (a);
```

a is the format code upon which PL/I will act. There are many types of formats available in PL/I, but before we go into those, let's look at precision and scale and what they mean to edited I/O.

Precision and scale

In every PL/I textbook I've read, the definitions of these two terms are thick as mud for beginning PL/I programmers, even in my favorite PL/I text by Joan K. Hughes (*PL/I Structured Programming*, Wiley and Sons, 1979). On page 108 the precision attribute is described in these terms: "The precision attribute specifies the number of significant digits of data and/or the decimal or binary point alignment. The precision of a variable is either attributed by default or it is declared along with the base and/or scale; it is never specified alone. It must follow either (or both) the base or scale in the declaration." And so on. Peter Abel (*Structured Programming in PL/I and PL/C*, Reston, 1981) comes out just as "thick" in his definition of scale on page 47: "The two scales are fixed-point (FIXED) and floating-point (FLOAT). In fixed-point scale (DECIMAL FIXED and BINARY FIXED), you specify the decimal or binary point, if any. In floating-point scale, the number is followed by an exponent that indicates the position of the assumed decimal or binary point." And so on.

These are two fine authors of excellent PL/I textbooks, so it's the subject that is difficult to explain in simple terms. For novice PL/I programmers, I have always preferred to refer them to the definitions in the old, original Digital Research PL/I-80 Applications Guide which states on page 137: "Decimal variables and constants in PL/I-80 have both *precision* and *scale*. Precision denotes the number of digits in the variable or constant, while scale defines the number of digits in the fractional part. For Fixed Decimal variables and constants, the precision must not exceed 15 and the scale must not exceed the precision. The precision and scale of a PL/I-80 variable is defined in the variable's declaration:

```
declare x fixed decimal (10,3);
```

while the precision and scale of a constant are derived by counting the number of digits in the constant, and the number of digits following the decimal point. The constant

```
-324.76
```

for example, has precision 5 and scale 2."

Because we are looking at PL/I in an informal way, let's get downright familiar in our approach to precision and scale. For the sake of simplicity, precision will sometimes be referred to as *width*, the total width of the numeric field. Scale will sometimes be referred to as *decimal places*, places to the right of the decimal point.

In PL/I-80 the maximum single precision is 24 (base 2) with double precision of up to 53. Scale cannot, of course, exceed precision, and it can equal it only if there are no places to the left of the decimal point. Precision for type float and type fixed binary is always given in binary (base 2) because it doesn't work easily in decimal (base 10). Two to the 24th for example is 16777216 (base 10) giving a precision of "7 to 8" which doesn't sound as precise as "a binary precision of 24." Type fixed decimal or binary coded decimal has a maximum precision of 15. Type fixed decimal is used primarily for calculations where no binary round-off error can be tolerated, as in accounting applications.

Here are the figures for the maximum precision allowed for the following data types:

Fixed (binary)	15
Fixed decimal	15
Float (binary)	24 single precision
Float (binary)	53 double precision
Character	254
Bit	16

Format codes

Referred to as "data format items" in the Digital Research manual, I like to call them format codes, and they are A, F, E and B. Before diving right into these format codes and what they signify, here is some general discussion. First of all, beware of confusing data types and data representations. The format codes refer to data representations. Data types are another animal altogether.

Let's review the PL/I data types used in edited I/O:

Fixed Decimal
Fixed Binary
Float Binary
Character
Bit

►

The full set of PL/I also has the data type float decimal which is absent in Digital's subset.

Data representations fall into the following four categories:

Alpha or character data
Fixed point data
Exponential (scientific notation) data
Binary bit data

Each one of the format codes corresponds to one of the four types of data representation.

A Format — Alpha data
F Format — Fixed point data
E Format — Exponential data
B Format — Binary bit data

Now let's back up a bit. Everything coming and going, in other words all data input and output, comes and goes as character. This is one of PL/I's tall hats, and in this way it is just like the C programming language. Why this is beneficial is best left for another chapter, but for now just remember that all data I/O is character.

Edited I/O deals with ways to represent these data on input or output. If you are outputting character (alpha) data, it's a snap. All you have to do is determine the field width or let it default to 254. The A format code is used for alpha data representation.

With numeric I/O, you can specify the F or E format codes. F is for fixed point data representation, E for exponential. If you use the F format with no scale, it'll output an integer-type number with no decimal places. If, on the other hand, you specify precision AND scale with the F format, you will get true fixed point representation. The E format causes I/O to be represented as a power of ten (scientific notation), which of course is the way to represent Carl Sagan's "billions and billions" without cluttering up the screen with zeros!

The B format code is for binary bit representation. Used by programmers in the "expert" class, binary bit representation is not for everyone. Let me note here, however, that PL/I is one of the few major programming languages to deal with I/O in the true binary format of 1's and 0's.

With that as a general overview of format codes, let's examine each code separately.

F Format

The F format code is used for fixed point arithmetic data on both input and output. It has the following form:

F (width, decimal_places) or F (precision, scale)

The number of characters specified by *width* is read on input and printed on output with the F format. The maximum precision is 15 places. Leading and trailing blanks are ignored.

The scale for type fixed decimal is optional with the F format, and if not specified is zero. For example, F(15,2) signifies a fixed point number with 15 digits (with a decimal fraction), and it outputs numbers like

12345678901 23.45

F(6) signifies a fixed point number with 6 digits (no decimal fraction) and it outputs numbers like

123456

E Format

The E format code is used for scientific notation or exponential data representation. On input, the E format converts the number input to type float binary. It has the following form:

E (width, decimal_places) or E (precision, scale)

Like the F format, *width* specifies the total field width, and *decimal_places* the number of decimal places in the decimal fraction. You've got to watch the E format on output because there is a "Catch 22." The *width* amount must be seven larger than the *decimal_places* amount to allow a place for the sign on both the number proper and the exponent "E" (not to be confused with the E format), a leading blank if no sign is present, and a decimal point. Typical E format output looks like

-1.23456E+05

The maximum precision for type float binary is 24 for single precision, 53 for double precision. (DRI's PL/I-80 version 1.3 and earlier did not have double precision.)

A Format

The A format code is used for the purpose of outputting character data, and it has the following form:

A (width)

Here *width* specifies the maximum width of the character field. (The A format code can be omitted in PL/I-80 and PL/I-86, but it is mandatory in the full set of PL/I.)

On input, the input field will be read up to, but not including, the carriage return if *width* is specified. On output the full string will be output if *width is omitted.*

Here comes the good news and the bad news. When width is specified on input, the field will be truncated if the field exceeds the *width* amount. If the field is less than the specified *width* amount, it will be padded to the right with blanks.

Here are some examples of program statements using the A format code:

```
get edit (str) (a)     /*reads anything up to 254 characters*/
put edit (str) (a)     /*writes anything up to 254*/
                         *characters                          */
get edit (str) (a(6)) /*reads up to 6 characters of the      */
                      /*input field and pads with blanks     */
                      /*if necessary                         */
put edit (str) (a(6))/*writes up to 6 characters             */
```

B Format

The B format code is used with bit strings. The B format has the following form:

B ( )

Only the characters 1 and 0 are used in I/O. The number of bits used for each digit is specified within the parenthesis:

B (8)

On output data is converted to bit strings and then to character. If necessary bit strings are padded with blanks to the right.

Control formats

Referred to as "Control Format Items" in the Digital Research manual, I like to call them control formats. Control formats are used to control what we associate with "white

space" such as tabs, spaces, carriage returns, line feeds and page ejects. Creative use of control formats is how you get all that neat and tidy spacing and column work associated with professional looking output.

Column

The column control format has the following form:

column (n) or col (n)

Column (n) moves the format pointer to the column specified by *n*. On input it causes the characters preceding the nth column to be ignored (a carry-over from card reading days). On output it starts printing at the nth column.

If the format pointer is past the point on the line specified by *n*, it goes to the nth column on the next line. For example

```
put edit (name) (col(40),a));
```

will print *name* starting at the 40th column (the middle of the screen). As its name implies, *column* is ideally suited for printing columns used so much in business and scientific output. The column location *n* can be an expression, a constant, or a variable.

Skip

The form for skip looks like this:

skip (n)

which causes a line feed or *n* line feeds to be printed on output.

On input it will wait for *n* line feeds before going to the next format item. On input a *skip(1)* is assumed if no skip control format is specified.

Let's play with the last two control formats. Providing the screen is clear to start with, the following program fragment

```
put edit(
  a demonstration,
  '',
  of formatted PL/I,
  '',
output)
  (skip(20),5(skip, col(32),a));
```

will print our message starting at the 21st line and the 32nd column, about the middle of a conventional screen. There is a great deal of power and flexiblity in this technique. It allows the user to format the output without the inconvenience of repetitive *put list* or *put edit* statements. The key is the comma at the end of each string constant, most clearly demonstrated with the null strings". This one takes some practice to get used to, but it is well worth the effort.

X

The X control format is a familiar format item to FORTRAN programmers. The form looks like this:

X (n)

The above advances the output *n* spaces. For example, the program statement

```
put edit (month, day, year) (f(2),X(6),f(2),X(9),f(4));
```

prints a two digit month, skips six spaces, prints a two digit day, skips nine spaces and prints a four digit year. It can be simplified with the repeat factor:

```
PUT EDIT (month, day, year) (3(f(4),X(6)));
```

This prints the variables *month, day* and *year* (here up to four digits apiece) all followed by six spaces. The variations are infinite.

Line

Line is only for print files. There are as many rules for the line control format as an NFL rule book, but we're just going to take a quick look at it. Here's the form:

line (n)

Here *n* specifies the next line to be printed. If more than *n* lines have already been printed on that page, you have an endpage condition. What happens then is determined by how you decide to handle this situation programmatically.

Page

Page is also used for print files only. It causes a page (↑L) or form feed to be written. It form feeds a new sheet of paper to the tear line. It also sets the line number and column to 1.

Remote format

For people who are fresh in from FORTRAN and who can't get used to putting the format on the same line as the variables and constants, there is remote formatting. The form

r (format_label)

indicates that the format specification is remote (located somewhere else). For example

```
put edit (month, day, year) (format_1);
```

means that somewhere there is a format statement preceded by a label, like this one:

```
format_1: format (A(8),X(2),F(2),X(2),F(4));
```

The word "format" is mandatory.

When formats are repeated frequently, this can be a real time saver.

Picture format

Version 1.3 of PL/I-80 heralded the inclusion of the picture format to edited output. PL/I's picture format is like COBOL's *picture* (and perhaps COBOL's only saving grace), and it resembles BASIC's *print using* attribute.

Where exact control of the output format is critical for neatness and clarity, the picture format is invoked. The following is typical of a picture format output:

$123,456,789.01CR

You have many options available with the picture format, and here is a list of them:

static/drifting characters	$ + − S
conditional digit characters	* Z
digit characters	9
decimal point position character	V
insertion characters	/ : , . B
credit and debit characters	CR DB

The syntactic rules for picture are not difficult to master. To invoke the picture format, use the letter "P" followed by the picture specification, as follows:

put edit (*amount* , amt) (p$z,zzz,zz9.99);

Let's take a closer at the options available with the picture format.

Picture Semantics and Static/Drifting Characters

Static/drifting characters *$*, *S* + and − require a bit of definition. The sign signs *S* '+' and ' −' will give either sign if *S* is used, '+' only if the output is positive, and '−' only if the output is negative. The *$* will always print out as a dollar sign. They are static if they appear only once, otherwise they are drifting.

data	picture	output
12345678.90	$SSSSSSSSV.99	$+12345678.90

Note that in the example, *$* is static, appearing only once. *S* (sign) is drifting, appearing all the way to the decimal.

Conditional Characters

The conditional characters *Z* and * are zero suppression characters. The asterisk will print wherever a zero or blank would have printed. The *Z* will substitute blanks.

data	picture	output
1.23	**********V.99S	********1.23+
1.23	ZZZZZZZZZZV.99S	1.23+

Insertion Characters

The letter "B," the comma, the period, and the backslash are insertion characters (*B* " , " " . " and " / ").

The insertion character *B* results in a space (blank). A blank will be inserted wherever a *B* appears in the picture format unless it is in a field of zero suppression characters that are suppressing zeros at the time.

data	picture	output
1234.56	z,zzz,zzzv.99	1,234.56
1101111	9999b9999	1101 111
011335	**/**/**	*1/13/35

9

The character *9* is in a class by itself, but it is closely related to the insertion character. It is an unconditional digit position. Wherever it appears, a digit will be printed (if one is available). All digits 1 thru 0 are printed.

data	picture	output
123456890	999999999v.99	12345678.90

V

Put as simply as possible, *V* shows where the decimal point is going to appear. Heed this warning: If it is not used, the system thinks you don't want a decimal point, and it won't give one.

Once *V* is encountered, suppression is turned off (suppressed!). All characters beyond it are printed.

data	picture	output
.33	9.v99	33
.75	9v.99	.75

Note how the first example turned the point off, not on.

DB and CR

DB and CR are debit and credit specifiers. They are very handy for use in accounting programs. If the data is negative and the CR/DB pair is in effect, here's what happens:

data	picture	output
1234567.89	$999,999,999.v**cr	$ 1,234,567.89
−1234567.89	$999,999,999.v**cr	$ 1,234,567.89cr

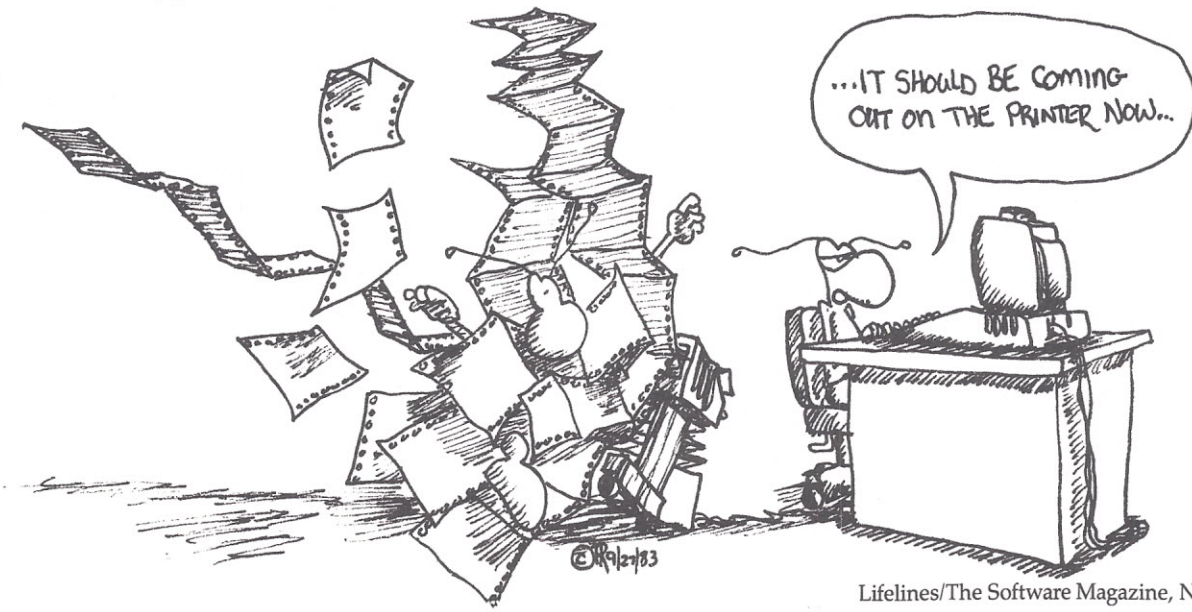
If you're into writing accounting packages (and I am), this feature is invaluable since it eliminates the need for debit and credit columns when the output is usually predictable. For example, entries in the sales credit column are credited, but in the special case of a memorandum specifying a debit, 'db' would clearly show it was an exception.

A few odd rules

If numeric data characters are zero and the picture does not contain a picture character *9*, the output will be asterisks if the picture character '*' is present, else the output will be blank.

WARNING If no sign picture is present [S + − CR DB] and a negative output is encountered, the dreaded conversion error ERROR(1) will be signaled.

Insertion characters and the CR/DB pair have no effect on precision. Precision will be one less than the number of characters in the picture specification.



Meet ***Lifelines™/The Software Magazine™*** and join the serious microcomputer software users who turn to us every month for timely, important software news.


Engineers,

executives,

professionals

and business people read our reports on newly-discovered bugs, patches to overcome bugs

or enhance software value, new products, and the changes in new versions. And these aren't the only ways we stretch our readers' software dollars: in-depth, comparative reviews really tell them which products are most suited to their needs.

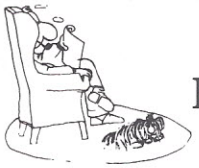

Programmers and

applications designers read ***Lifelines/The Software Magazine*** to stay in touch. Like the

computer and software dealers, the consultants,

and

professors who lean on ***Lifelines/The Software Magazine***, they have to know the latest on operating systems, programming tips and new trends in software — so their clients and customers are well-served.


Hobbyists keep up with The CP/M® Users Group and the largest library of public domain software in the world. If these people sound like you, join the software community. Subscribe to ***Lifelines/The Software Magazine***.

To order, please print below:

Name

Company

Address

City State Zip

U.S. Canada, Mexico
☐ 1 year, $24 (12 issues) ☐ 2 years, $46
Overseas
☐ 1 year, $50 (12 issues) ☐ 2 years, $98

☐ VISA ☐ MasterCard

Signature

Card Number

Orders must be prepaid; checks must be in $US, drawn on a U.S. bank.

Feature

MagicPrint Reviewed

by Walt Jung

MagicPrint by Computer EdiType Systems is a combination printer driver/formatter for letter quality printers which runs under CP/M-80 compatible operating systems. It can print files prepared on a variety of CP/M editors, including the most popular of these, WordStar. It is the WordStar version of MagicPrint that is reviewed here, version 1.22. A summary of the program's general essentials is listed in Table I.

Introduction

There are many text formatters available for CP/M, as well as many word processors which include built-in print capabilities. However, when it comes to full support of sophisticated printing tricks inherent in a precision printer (meaning a letter quality daisy-wheel, or a thimble type), the pack narrows considerably. Ironically, the most popular CP/M word processor, WordStar, does not fully support all of the features inherent in precision printers. Although such printers certainly can be driven from WordStar with generally good print appearance, the program nevertheless does lack a *true proportional* print feature. This is true for the older versions, as well as the more recent version 3.3.

Lest such statements be misunderstood by WordStar fans, true proportional printing includes not only variable character spacing for justification, but also goes further. A true proportional print wheel has characters which vary in width, between such extremes as the "W" and "i" letters, for example. Tables of space values for the characters, when stored within the program, are used to print text so that the wider characters are allotted more space than the narrow ones. The result is "true proportional," an aesthetically more pleasing form of print. For many people it is more attractive than standard fixed pitch, and can indeed approach typeset quality. Appearances aside, true proportional printing is also about 20% more efficient, in terms of text density.

Although WordStar has a "hidden" table for an undocumented proportional mode (invoked by ↑P, see customization notes), the results leave a lot to be desired. More importantly, since it is not a supported feature, it is extremely difficult to see what it should do, or how it might be best used.

MagicPrint is a program intended to address the problem of adding true proportional print capability, not only to editors such as WordStar, but any other CP/M editor or word processor lacking it. Since it is a standalone package, it will work with virtually any editor, including the popular ones. Special editor-optimized versions of the program exist not only for the WordStar version reviewed here, but also for several others (so specify the editor of interest, either when ordering or making an inquiry).

Operation

Since MagicPrint is a separate and independent program, it prints from already existing disc files. These may be either standard ASCII text files, or WordStar *document mode* files. High bits are no problem for MagicPrint, but it is stated that the *document mode* should be used for creating WordStar files for MagicPrint (true TABs are not allowed in non-document mode files, but the document mode multiple-space "TAB" is okay).

In using the WordStar version, I had no problems in printing various files either from CP/M, or from within WordStar from the "r" option of the OPENING or NO-FILE menu. In fact, the latter way of operating MagicPrint closely approaches the easiest way to use it, that is from within WordStar, after saving your file. Just make "MP FNAME.WS" your "r" option command. Overall, this would be a sequence consisting of edit, save, print, and check the printout. Then, reedit, and repeat it all again if necessary. You can, of course, use the program from the CP/M command level, with a similar syntax. In either case, MagicPrint prompts you for some options when it signs on, such as number of copies, page to start, etc., then proceeds with printing.

This sequence implies a difference between editing for printing with just WordStar, and editing for printing with WordStar and MagicPrint. WordStar is well known for its "what-you-see-is-what-you-get" feature of on-screen formatting. When editing files formatted for MagicPrint, that advantage is largely lost. The reason is that the text which appears on the WordStar screen often bears little direct resemblance to the printout, being peppered with formatting commands and special characters. By and large, this is simply a by-product of the separate formatter printing approach, not any inherent fault with MagicPrint (the same point would be true of other stand alone formatters as well).

However, to use MagicPrint most effectively, some degree of proficiency will need to be developed in predicting what a printout will look like. The only way to tell for sure is to just print it out. (You can expect to use a lot of paper, at least initially!)

Note: The above considerations are generally applicable to MagicPrint V1.22. As this review was being finalized, a new version was due to be introduced, with an on-screen preview feature. This addition should minimize a great deal of difficulty in using the program.—W.J.

The WordStar version of MagicPrint is set up for a minimum of differences between the standard WordStar imbedded control character commands, such as a ↑B pair for boldfacing, etc. These are recognized by MagicPrint just as they are used in WordStar, but additional commands are added, such as a ↑E pair for footnotes, ↑An (n=0-9) for variable pitch, ↑L for new page (instead of .pa), and ↑N for reset (instead of standard pitch).

In addition to the above imbedded characters (which incidentally assume a "[C" form in other versions), there are what is known as "comma commands" in MagicPrint. The usage is similar to a WordStar *dot* command (which are ignored by MagicPrint); that is, a command of the form ".C" is placed at the start of a line, and is terminated by a return (CR/LF pair). As far as the commands themselves, however, they are entirely different from those used in WordStar since they do so many different (and additional) things. Briefly, these are summarized in Table II.

New features compared to WordStar

MagicPrint has a couple of very nice features for general formatting, which themselves are either missing or not as fully realized in WordStar. While WordStar does handle headers and footers, MagicPrint allows multiple line titling with greatly enhanced flexibility. You can format headers and footers for anywhere on the page, and with pitch independent of the main text.

The footnoting addition is a most welcome enhancement, since this function is not available from within WordStar. In use, you simply place footnote keys [such as (1), etc.] where appropriate in the text, and define the footnote itself by entering text between a pair of ↑E's, placed anywhere. During printing, the footnote is placed at the page bottom, denoted by a solid line (see the print sample in Listing 1).

Installation

The program is supplied as a COM file "MAGIC.COM", which may be usable as is. The printer driver of this version uses the CP/M list device, and assumes that your operating system handles the protocol. In instances where this is the case (it is in the Heath CP/M used with my H89), you need do nothing further. For any given installation, you can verify whether the "as-supplied" MAGIC.COM is suitable, simply by installing a proportional wheel, and printing the supplied "SUMMARY" file.

Should you wish to customize MagicPrint, an INSTALL utility is supplied to do so. A sample log of my installation of a port driver is shown, in Listing 1. This session was captured via MicroShell (*Lifelines*, April 1983, p. 17). I find it very useful to save a copy of an installation sequence such as this, for future reference.

This printout serves as both an installation guide and a print sample for MagicPrint. Here, a driver is installed at base port E0, with an ETX/ACK handshake, driving a Diablo 630. Why the special driver, you ask? Simply because using the direct port driver for MagicPrint allows the operating system driver to drive a *second* printer at a different port, for drafts and general printouts directly from CP/M. This provides greater operational flexibility. By contrast, should you configure a letter quality printer with a proportional wheel to print directly from CP/M, it is not only a gross overkill for routine uses, but the listings will look decidedly odd!

For this sample, the "Old Proportional" setting of justification is used with MagicPrint, with an appropriate (fixed pitch) wheel. The appearance of this text is generally similar to what you might get from WordStar except, of course, that there is that nice footnote, courtesy of MagicPrint, which summarizes what it is all about. (Note that the answers to prompts supplied by the user are shown here in **bold face** print).

Printing True Proportional

Of course, the print sample contained in Listing 1 does not begin to show off the better print features of MagicPrint, since it is a fixed pitch printout, not true proportional. Now, to get into that neat stuff!

Listing 2 is a print sample which illustrates the quality of print attainable with MagicPrint, and shows the four different modes of justification printing the same text. This was produced by adapting the sample file, "SUMMARY", with minor adjustments for the context of this review (see footnote). A Diablo BOLD PS 96 WP wheel was used to print this sample.

The first two of the repeated four paragraphs illustrate the best of what MagicPrint can do, with true proportional print, both justified and unjustified. To compare the general appearance of these two paragraphs with non-proportional fixed pitch text (*à la* WordStar style), compare it with the text in the previous sample, setting aside the character style differences of the wheel. Note the distinct uniformity of appearance in the true proportional mode.

The last two paragraphs are intended to show the "Old Proportional" mode, but in one sense they exaggerate the difference. This is because they were printed with a proportional wheel, which will not give as uniform an appearance as a fixed pitch wheel (Listing 1). Unfortunately it is not possible to print in one pass with two such wheels (since the printer must be powered down and reset to change wheel types).

To supply a closer comparison of true proportional versus fixed pitch, these same two paragraphs were edited with WordStar, and adjusted for a printout similar in length to the first two paragraphs in Listing 2 (MagicPrint, true proportional). Printed from WordStar, with a fixed pitch wheel, they are shown in Listing 3. This printout gives more direct evidence of the true versus non-proportional differences of print. For example, compare the wide-open appearance of "old" in Listing 3, versus the true proportional counterpart in Listing 2. Note that justification tends to aggravate this factor, in Listing 3.

Summary

MagicPrint is a considerable enhancement to WordStar for those applications requiring the highest quality of printed text appearance. Beyond just appearance, however, it also offers the most welcome features of enhanced titling, footnoting and a generally expanded command set. On the minus side is the inability to see on screen the output (in V1.22; a follow up review on the new version with the preview feature is planned). Still, in my opinion this is substantially offset by the reward in both the final product appearance, and the additional features.

There are some word processors which by themselves can achieve the quality of print output represented by a WordStar and MagicPrint combination. One would be Spellbinder (see *Lifelines*, April 1983, p. 3, plus the follow-up addendum), a program which has macro features in addi-

tion, but does lack footnooting. However, for those already equipped with WordStar, MagicPrint is a good choice, as a minimum of new learning must be accommodated (as opposed to starting out fresh with an entirely new word processor). And, the variety of other support packages and enhancements available for WordStar speak strongly in its favor in general regards, as well. Indeed, along these very lines is a new *true proportional* merge-printing capable program from Computer Editype Systems, "MagicBind." Look for a review on this program in the future, along with the follow-up on the MagicPrint update.

To sum up, MagicPrint is a well done program, and also one which greatly enhances the operation of WordStar. The total of the two is a potent combination. While MagicPrint may not be trivial to master completely, it can result in very high quality printing, and should be considered if that is among your goals.

Table I **Facts and Figures**

Program Reviewed:
MagicPrint version 1.22, WordStar version. A precision printer driver/formatter, with true proportional printing capability.

Company:	Dealer:
Computer Editype Systems	Lifeboat Associates
509 Cathedral Parkway 10A	1651 Third Avenue
New York, NY 10025	New York, NY 10028

Price: $195

Operating Systems: CP/M-80 SB-80

Memory Requirements:
36K (min), 48K recommended

Editor Requirements:
A variety of CP/M-80 editors can be used. See text.

Printer Requirements:
A letter quality printer, of the daisy wheel or thimble type. Compatible with Diablo 630 and NEC Spinwriter.

Systems Supported:
Virtually any via CP/M LST driver with handshake (as supplied). Alternately, user installed custom I/O.

File Package:
MAGIC.COM program and INSTALL utility, plus SUMMARY sample demo file.

Documentation:
Approximately 50 pages including index, three-ring punched/stapled copy. Printed with MagicPrint.

Table II **MagicPrint Comma Commands**

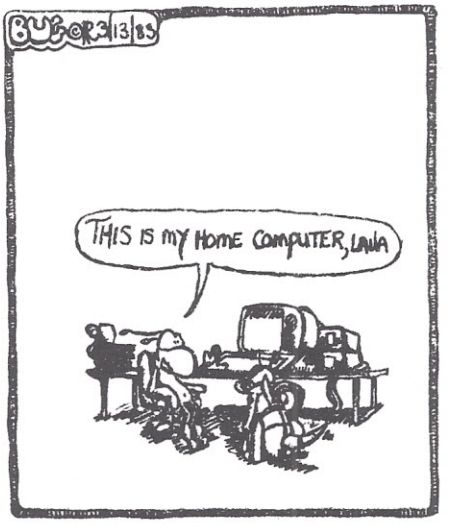
,B	Moves backwards
,C	Centers line(s)
,D	Restores defaults
,E	Right flushes partial lines
,F	Pairs page
,G	Sets page length
,H	Change character density
,I	Indents
,J	Justify (one of 4 modes, see samples)
,L	Sets line length
,M	Sets left margin
,N	Same as ↑L, or conditional
,O	Outdents
,P	Set page number
,R	Rightflush line
,S	Sets spacing
,U	Sets broken/solid underscore
,V	Sets line height
,W	Right indent
,RETURN	Pause print (same as ↑C)
,-	Page number
,*	Page titling
)	Ends formatting block

Listing 1

INSTALL PROGRAM for MagicPrint Ver. 1.22

```
Disk Drive where MAGIC.COM is located
       (Do not press RETURN)
A, B, C, D ... or P: b

Choose a number that corresponds to your computer system and printer port number (*)
as shown in parentheses.
* denotes systems on which MagicPrint
has been tested.

If none of the numbers matches your system,
type 0 (zero) and answer the questions
to customize your printer I/O.

Typing errors will redisplay the beginning;

 A -  Altair 88-2SIO (12, 13)
 E -  Elec Control Tech R2-I/O (0, 1)
 I -  IMSAI SIO (3, 2)
 J -  IMSAI MIO (33, 32)
 S -  Super Brain Auxiliary Port* (40, 41)
 T -  Super Brain Main Port (58, 59)
 X -  Xerox 820 (05, 07)*

 Press RETURN for next display;
                        Control-C to abort
 ANSWER: RETURN

 C -  Cromemco TUART (0, 1)
 I -  IMS (10, 11)
 N -  North Star left serial port (2, 3)
 O -  North Star right serial port (4, 5)
 S -  SDS Systems (7D, 7C)*
 T -  TRS-80 Model II Port A (F4,  F6)*
 U -  TRS-80 Model II Port B (F5,  F7)*

  Type O for Custom I/O; R to restart; RETURN to skip
  ANSWER: O
  You may choose one of three ways to specify Printer I/O:
  If you wish to use your own printer I/O, type X
  In order to use CP/M List Out routine, type Z
  In order to customize the program, please answer
  the following questions with an appropriate number
  in hexadecimal - (05, 5A, 7F, etc)
  Chapter 6 of the manual gives a more
  detailed explanation about the questions.
  press RETURN to skip
  Type X (your own routine), Z (CP/M List Out), or
  1. Your printer INPUT STATUS PORT number: E5
  2. The MASK byte for this port (02, 80, etc): 01
  3. Bit(s) set when data is available: 01
  4. Your printer INPUT DATA PORT number: E0
  5. Your printer OUTPUT STATUS PORT number: E5
  6. The MASK byte for this port (01, 80, etc): 20
  7. Bit(s) set when data is available: 20
  8. Your printer OUTPUT DATA PORT number: E0
  Is everything correct?
  Type Y to proceed; CONTROL-C TO ABORT
  or else return to Question 1: Y
  Choose a number that corresponds to
  the printer you use
```

```
D - DIABLO 1610/20*
E - DIABLO 630 (XEROX 1730)*
F - DIABLO 1640/50 (XEROX 1740/50)*
Q - QUME SPRINT 5 and 9 Series
S - C.ITOH STARWRITER, F-10, VISTA VERSION
press RETURN to skip E
Select which hand-shaking signals your computer uses:

1.  ETX/ACK (03, 06)
2.  XOFF/XON or DC3/DC1 (13, 10)
3.  No handshaking necessary (using 300 baud, etc
RETURN to skip  1
                JOB COMPLETE!
```



(*) The installation example is for the Heath H/Z-89, with a Diablo model 630 1200 baud printer at port E0, with ETX/ACK handshake. This printer can also be driven with the pre-configured driver in **MagicPrint**, but use of the port driver gives greater flexibility (see text). This print sample is in Old Proportional Justified, with a 12 pitch wheel.

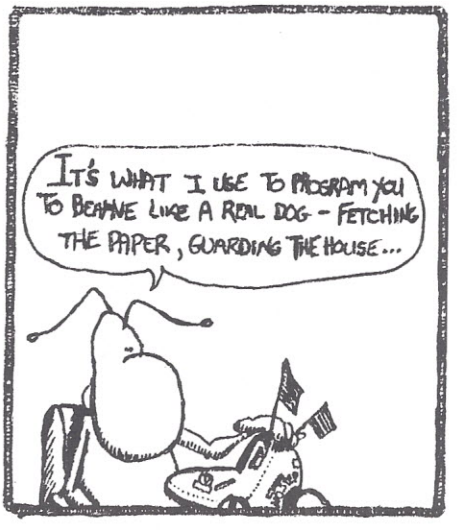
Listing 2

An Overview: (1)

The word processing gear has been improving steadily for years. Due to shortcomings in the printing process, however, the final copies produced by the expensive equipment have not shown improvement over what comes out of an electric typewriter.

MagicPrint by CES remedies this deficiency by way of "true proportional spacing" that allocates character spacing according to the width of individual characters, For example a W or an M gets two and half times as much space as an i. The printed pages thus produced look as if they had been done by a professional typesetting shop,

MagicPrint is fully bidirectional even when printing embellished characters such as **bold face** or ~~overstrike~~. The last character of a line is printed at stationary position before hopping over to the next line. These features improve printing speed and minimize carriage movements and equipment wear.



There are four modes of printing under this program: true proportional spacing justified and unjustified; the old proportional spacing (the conventional mode - giving equal width to all characters) justified and unjustified. **Default** is true justified. This paragraph is typed four times to show the four different modes of print out. (This paragraph in **True Proportional Unjustified**).

There are four modes of printing under this program: true proportional spacing justified and unjustified; the old proportional spacing (the conventional mode - giving equal width to all characters) justified and unjustified. **Default** is true justified. This paragraph is typed four times to show the four different modes of print out. (This paragraph in **True Proportional Justified**).

There are four modes of printing under this program: true proportional spacing justified and unjustified; the old proportional spacing (the conventional mode - giving equal width to all characters) justified and unjustified. **Default** is true justified. This paragraph is typed four times to show the four different modes of print out. (This paragraph in **Old Proportional Unjustified**).

There are four modes of printing under this program: true proportional spacing justified and unjustified; the old proportional spacing (the conventional mode - giving equal width to all characters) justified and unjustified. **Default** is true justified. This paragraph is typed four times to show the four different modes of print out. (This paragraph in **Old Proportional Justified**).

(1) This print sample was adapted from the supplied "SUMMARY" demo file, which is (C) and property of **Computer EdiType Systems.**

Listing 3

There are four modes of printing under this program: true proportional spacing justified and unjustified; the old proportional spacing (the conventional mode - giving equal width to all characters) justified and unjustified. **Default** is true justified. This paragraph is typed four times to show the four different modes of print out. (This paragraph is **Unjustified**, printed from WordStar with a 10 pitch wheel).

There are four modes of printing under this program: true proportional spacing justified and unjustified; the old proportional spacing (the conventional mode - giving equal width to all characters) justified and unjustified. **Default** is true justified. This paragraph is typed four times to show the four different modes of print out. (This paragraph is **Justified**, printed from WordStar with a 10 pitch wheel).

The CP/M Users Group

Volume 93

CP/M Users Group
1651 Third Ave. New York, NY 10028
(212) 860-0300, Ext.343

DESCRIPTION: North Star Double Density/Jade DD Combined BIOS
North Star Double Density BIOS with PMMI modem and 80-track drive support
North Star Double Density full disk copy and format program
North Star Double Density sysgen program
North Star Single Density BIOS, BOOT, and sysgen program
Jade DD format program and on-board controller (DCM) program

by Steven B. Bogolub

NUMBER	SIZE	NAME	COMMENTS
	2K	-CPMUG.82A	CONTENTS OF CP/M VOL. 82A
	8K	ABSTRACT.82A	Volume abstract
	1K	FILES.CRC	CRC of files on this disk
	2K	CRCK.COM	CRC program.
	12K	USQ-19.COM	Unsqueeze utility (to unpack *.?q? files)
	5K	U-G-FORM.LIB	CPMUG submission form
82A.1	27K	FORMAT.AQM	Source for Jade DD FORMAT program
82A.2	7K	FORMAT.COM	Object of Jade DD FORMAT program (requires running NBIOS82A system)
82A.3	34K	MBIOS56.AQM	North Star Double Density BIOS with 80-track drive and PMMI MM-103 modem support
82A.4	1K	MBYE.ASM	Trivial BYE program to hang up phone on MBIOS
82A.5	36K	NBIOS56.AQM	North Star Double Density BIOS with Jade DD (8″ drive) support
82A.6	22K	NDCM2.ZQ0	Jade DD on-board Z80 controller code
82A.7	20K	NSBIOS.DQC	Notes on program contents and implementation
82A.8	26K	NSCOPY.AQM	North Star Double Density Copy and Format program source (has own disk drivers)
82A.9	4K	NSCOPY.COM	North Star Double Density Copy and Format program object
82A.10	8K	NSGEN.AQM	Source for sysgen utility, to write NBIOS82A or MBIOS82A system to disk
82A.11	1K	NSGEN.COM	Object for sysgen utility (double density)
82A.12	14K	SDNBIOS.AQM	North Star Single Density BIOS
82A.13	2K	SDNBOOT.ASM	North Star Single Density BOOT
82A.14	8K	SDNSGEN.AQM	Source for sysgen utility, to write SDNBIOS system to disk
82A.15	1K	SDNSGEN.COM	Object for sysgen utility (single density)

Prepared by Steve Bogolub 07/83

This volume consists entirely of Steve Bogolub's North Star CP/M version 2.2 BIOS routines. They are copyrighted, but available freely for non-commercial use. Extensions of programs formerly included on Jade's CP/M 2.2 release disk are included with special permission from Jade Computer Products, Inc., for use with the Jade Double D Rev C disk controller.

This volume is an update of CPMUG Volume 82, providing enhanced versions of NBIOS56 and NSCOPY, and extra support programs for the Jade Double D. The Volume 82 version of the single density BIOS had fatal bugs in the disk write routines, as well as being tailored for a specific machine with many non-standard features. The new version of the single density BIOS supplied on this disk corrects the problems, and will run on a much more standard machine (North Star Horizon). All source files can be assembled with the standard Digital Research assembler ASM, except for the Jade support programs (FORMAT.AQM and NDCM2.ZQ0, see below).

To provide more in the same amount of release disk space, most of the source files are supplied in "squeezed" format. To restore to their normal source format, use the unsqueeze utility provided. For example, to unsqueeze NBIOS56.AQM from the D: drive to the C: drive, log into the D: drive and type the following:

D>USQ-19 C: NBIOS56.AQM

This assumes the release disk is loaded in the D: drive, and a disk with approximately ⅓ more space than the squeezed file occupies is loaded in the C: drive, and will produce the unsqueezed file NBIOS56.ASM on the C: drive.

Included are the following files:

USQ-19.COM

The above-mentioned unsqueeze utility, written by Richard Greenlaw (another copyrighted public-domain program).

NBIOS56.AQM [squeezed source file]

A BIOS for North Star MDS-AD2 double density 5.25″ floppy disk units 1 and 2 as CP/M drives A: and B: and a Jade Double D Rev C 8″ floppy disk units 0 and 1 as CP/M drives C: and D:. This BIOS supports both the standard North Star controller base address (0E800H) and the normal relocated base address (0FC00H) using information supplied by the on-board PROM boot routines, allowing full utilization of the memory in a relocated system. Disk formats are compatible with two Lifeboat single-sided versions (1K and 2K group sizes). Only single-sided 35-track drives are supported. Console I/O supports a QT IO+ board, but for convenience, checks for this board at the genned-in address, and automatically reverts to a normal North Star Horizon config if the IO+ does not re-

spond (console on left serial port, printer on right serial port with DTR handshake). If the Jade controller does not respond to the configured port address, it will be ignored. Therefore, a standard North Star Horizon will be supported by the same object program as the enhanced system with the IO+ and the Double D, or a relocated controller.

NSCOPY.AQM [squeezed source file], NSCOPY.COM
This is a utility to allow the operator to copy, format, or validate North Star single or double density disks, using the MDS-AD2 double-density controller. Copying can be done between any two North Star drives, and can include all tracks, or just the system (0 and 1) tracks or data tracks (2 - n). This program is initiated under CP/M, and uses the active BIOS console routines, but has its own disk drivers, so can be run (for example) under Lifeboat CP/M. If a relocated controller is present in the machine at 0FC00H, it is automatically used, otherwise the standard controller at 0E800H is required. All four units can be specified, and formats are defined for 40-track single-sided and 80-track double-sided drives (for use with MBIOS56), in addition to the normal 35-track single-sided drives.

NSGEN.AQM [squeezed source file], NSGEN.COM
Provides the functions of the Digital Research standard CP/M 2.2 program 'SYSGEN', but for the NBIOS56 and MBIOS56 versions of CP/M 2.2.

MBIOS56.AQM [squeezed source file]
This BIOS is specialized for remote modem dialup use. It incorporates elements of Dave Jaffe's BYE program. If the memory location "MODEM" contains a value of zero when the coldboot routine "INIT" is entered, a special set of console vectors is patched over the genned-in set to handle the PMMI modem instead of the normal Horizon drivers. Variable baud rate and other features are supported. This is a North Star-only BIOS (no 8" Jade), but units 0 and 1 are defined to be 40-track single-sided drives, and units 2 and 3 are defined to be 80-track double-sided drives. MBIOS56 does NOT relocate the controller base address at runtime (like NBIOS56 does), but the controller equates can be changed for a relocated controller with no problems.

MBYE.ASM
Trivial BYE program that can be run when dialed into the MBIOS system to "sign off" instead of just hanging up. Merely says "Good-bye," then jumps to MBIOS cold boot routine.

SDNBIOS.AQM [squeezed source file]
BIOS containing routines to support a North Star MDC-A4 single density 5.25" floppy disk controller, with units 1 and 2 as CP/M drives A: and B:, using standard Horizon left serial port as console and right serial port as DTR-handshake printer. To improve access times (as in make tolerable), full track buffering is done, which requires an extra 2.5K of memory, so MOVCPM ends up needing to be run for 3K less than the desired system size. Disk format is compatible with Lifeboat CP/M version 1.4 disks, but is NOT set up as a dummy 8" floppy. Format is also compatible with NBIOS56 and MBIOS56 single density. This BIOS should REPLACE SDNBIOS and SDNUSER from CPMUG Volume 82, as the disk deblocking routines in the older version had severe (and highly destructive) bugs in the disk write routine, which were mistakenly blamed on the small size (21K, one disk drive) of the original machine. The new routines were tested on a standard Horizon with 24K of memory and two drives.

SDNSGEN.AQM [squeezed source file], SDNSGEN.COM
Provides the functions of the Digital Research standard CP/M 2.2 program 'SYSGEN', but for the SDNBIOS version of CP/M 2.2.

SDNBOOT.ASM
This routine is read off the disk from track zero, sector four by the North Star single density BOOT PROM. It is responsible for reading in the BIOS routines from the disk, then jumping to the BIOS cold boot entry point.

NSBIOS.DQC [squeezed DOC file]
Contains notes on all three BIOS implementations, and includes information on how to bootstrap one of these systems starting with a Lifeboat CP/M version 2.2 system as a starting point.

In addition, the two programs required to support the Jade Double D under NBIOS56 are included:

FORMAT.AQM [squeezed source file], FORMAT.COM
Enhanced version of the release Jade FORMAT program, providing the capability to format a wide variety of 8" double density disks for use under NBIOS56 (most notably 1024 bytes/sector), as well as standard single density. Requires NBIOS56 system to run, due to an additional needed BIOS vector. If changes to the source are required, Digital Research's MAC assembler is required.

NDCM2.ZQ0 [squeezed source file]
Enhanced version of the release Jade DCM2 on-board controller program, converted to Cromemco source format (Zilog mnemonics). Used with NBIOS56 system to support the Jade Double D.

Feature

The E-COM Connection

by Jon Wettingfeld

Electronic Mail: Something to write home about

The goal of using electronic systems to improve business organization communication and decision making will certainly be closer to realization through implementation of the new electronic mail and message systems. New communication networks utilizing techniques such as facsimile reproduction, communicating word processors, upgraded communications switching in Telex and TWX nets are already having a major impact on office efficiency. Farther in the future, many sources agree, widespread use of digitized voice messages transmitted to "electronic mailboxes" will reduce both paper costs and delays in telephone contact. Additionally, teleconferencing with both moving and static images will significantly reduce travel time and expense. While all this will not occur immediately (for the near future, most electronic mail is likely to be paper based, paperless systems not being widely available until the 1990's), there are already many options in computer based mail systems (CBMS). One of them, E-COM, the U.S. Postal Service's electronic mail, is notable for its cost effectiveness, flexibility, and because it is available to virtually every personal computer user. We'll discuss the service in more detail below.

Other systems in use at the present time include "Access" which, along with TWX and Telex, is available through Western Union, "subscription" CBMS's like EMS in Washington, D.C., Compuserve and Dialcom, and campus-based mail systems such as Omnicon. FAX, a facsimile transmission system, is being used very cost effectively by many groups including NASA.

One important CBMS which has been available for quite some time is the communicating word processor. Word processors, which were originally used on a standalone basis, are now frequently employed in local area networks (LANS). For example, NBI Corporation reports that 30% of its word processing machine sales include a communication option for tying into such networks. Meanwhile, where incompatibility between machines by different manufacturers (and the software packages that run on them) may be an obstacle to networking, code and protocol translators are becoming available to alleviate some of these problems.

Another important vehicle for computer based mail is an already existing LANS installed by various hardware manufacturers for local and regional communication. Some of these include "Wangnet" (Wang Labs), Ethernet (Xerox), Cablenet (Amdex), and Localnet (Sestek). All are well suited for electronic mail use, with software readily available for transmitting it.

As mentioned before, Western Union's "Access" is also a significant option in the electronic mail market. The system allows almost any hard copy business communications device to connect to any other at a low cost. This is done via a regular telephone line hookup to "Infomaster," one of the world's largest communications complexes. Here also, users can convert their electronic messages to many paper-based message services: same day telegram, international cablegram, mailgram, etc., with the resulting advantage of flexible delivery times.

Other "subscription" electronic mail services of note include the "Comet" CBMS (which is a "turnkey" communications system used by companies like Manufacturer's Hanover Trust), and Compuserve and Dialcom. Compuserve and Dialcom, incidentally, are allowing some members of Congress to communicate regularly with hundreds of constituents as well as congressional advisors on a weekly basis. Most of these services are available to regular microcomputer users. But one of the most sweeping innovations in electronic mail to date has come from the U.S. Postal Service, which has brought electronic mail to an even wider public.

E-COM: Neither rain nor parity nor protocol...

E-Com, the U.S. Postal Service's version of electronic mail, was originally a controversial proposal. Many in private industry, from small printing shops to those engaged in other CBMS-type services, felt — and still feel — that the operation represents unfair competition, which is subsidized by other types of U.S. mail service. However, the Post Office has been involved with such enterprises for quite a few years. It operated what might loosely be called one of the earliest prototypes of the electronic mail system — the telegraph line between Baltimore and Washington D.C., which went into operation in 1845! Later, briefly, it operated all of the telephone and telegraph service in the U.S. during the First World War. In both cases, service reverted to private ownership in a short time.

More recent efforts towards starting a separate Postal Service electronic mail system include two facsimile services, one of which began in 1959. It was called "Speed Mail." But a more permanent and definitive step was taken with the advent of the mailgram in 1970. With Western Union controlling, managing, and marketing the service, the Post Office began cooperating with the "common carrier" to provide delivery nationwide. Currently, the system allows the user to send his message on magnetic tape or disk, via telex or twx terminals, or by simply telephoning Western Union. The message is then transmitted to the Post Office and mailed for delivery on the next business day.

E-Com, the newest service, was begun in January 1982. It uses a com-

mon carrier - the Western Union "Infomaster" computer, mentioned above. While still paper based, it permits a wide range of 8-bit and some 16-bit micros, along with other terminal types, to communicate with the U.S.P.S. Messages generated on such machines are transmitted electronically via phone lines to 25 service postal centers. Equipment in the postal centers prints the messages and puts them in distinctive blue and white envelopes. Messages are then delivered in two business days via regular mail service.

Though initial optimism about projected traffic in E-Com has yet to be borne out, several factors suggest that the service could come into wider use. One is simply the fact that an estimated 30% of all first class mail is generated with computer assistance. This 30% could be readily linked to and carried by the E-Com system. These factors include the cost effectiveness of the system and the fact that software and hardware, which allow the micro end user to utilize the system, are and have been widely available almost since its inception.

A good example of a software package meeting Post Office requirements for becoming a "certified" E-Com "carrier" is Digisoft Computer's Mail Com. Carried by Lifeboat Associates and other software houses, it is available for both IBM-PC (and XT) in 16-bit, and for most 8-bit CPM-80 machines. (The 8-bit version includes menu driven installation for most common terminals, and a customizing option for those terminals not on the list.)

The package comes with an editor which allows the user to create and edit both address files and text. Of course, the user can employ any word processing program to do this, as long as control codes not recognized by the Postal Service are omitted (such as the "dot" commands used by WordStar, etc.). The package allows the user to store text for between 450 and 600 different letters per 100K disk space, and allows transmission of 200 or more messages per communication (the current minimum per transmission is 200).

To utilize E-Com, one must additionally have a modem (acoustic or non-acoustic) and a communications package to allow your micro to talk to the modem. The user also has to register with the post office to become a certified carrier. This involves mailing an application (which is included with Mail Com's documentation) and then running tests with the software to ascertain that transmissions are being formatted correctly. When this is done the fifty dollar yearly fee is paid, and certification is complete. Subsequent costs include 26 cents per one-page letter, and 31 cents for the maximum of two pages. The minimun transmission quantity rule is currently under review, and may be changed to attract more customers). Each user is assigned a unique identification number which is sent with each transmission, and must also give an estimated cost for each group of letters sent.

E-Com utilizes four distinct types of messages, and Mail Com of course has capabilties for each. The messages are called Single Address Messages (SAMs), Primary Text Transmissions (PTX), Common Text Messages (COTs) and Text Insertion Messages (TIMs). These different types allow great flexibility in the letter formats.

Briefly, they can be described in the following way. The SAM is the simplest kind of message. It is most like the traditional letter. The transmission is a unique text intended for only one recipient with no variable or inserted information.

Common Text Messages let you send identical messages to a large number of recipients *à la* a form letter. COTs may only be used in this way, without employing any variables other than the addresses and addressee names.

The TIM is utilized in conjunction with the PTX to produce letters with similar content but having variable insertions of text. The PTX serves as a skeleton into which TIMs are inserted according to special text insertion codes specified by the user.

Obviously these message types provide great flexibility, ranging from "boiler plating" and form letters to individual mailings. In addition, variable margin settings, tab stops, and special postal "Box Codes" allow invoices and bills to be formatted and transmitted directly to the E-Com sites. While, as pointed out, delivery time is usually two days, the Post Office suggests that time should be saved because of the direct electronic transfer from the computer. Additionally, a useful aspect of the system is that E-Com will keep your PTX on file for a 24-hour period. This means that you can send several transmissions of TIM messages in this period that will utilize the original text "skeleton." They will automatically be merged with the original PTX messages where the insertion codes indicate.

Mail Com itself comes with built-in error trapping. Problems arising from use of too many special options per line (e.g., Boxes) are flagged, as are cases where the number of text insertions don't agree with the number of insertion codes in the current address file. Warnings are also issued if correct information or address diskettes are not in the drives, and if access and user I.D. codes are not recognized by E-Com. Additional safeguards include automatic backup when address or text files are being edited, assuring you that previous work will be protected if any problems develop.

In short, with a package like Mail Com, E-Com can be utilized for many functions, especially for such things as direct mail advertising and billing and credit notices. While acceptance has not been immediate or widespread, and there has been some discussion of a rate increase (which will not be in effect until at least February), there is little doubt that it can fill a significant commercial need for mass mailings with potential savings. In the future, innovations in the service may include laser printers to imprint black and white corporate logos on letters, and inclusion of reply envelopes for billing and fund raising.

Send Us Your Bugs

We are actively searching for bugs, bug fixes, and patches for software programs.

Write or call:

Editorial Dept.
Lifelines/The Software Magazine
1651 Third Ave.
New York, NY 10028
Telephone: (212) 722-1700

Feature

A Review of WordIndex II

by David Carroll

If you've ever battled with a lengthy writing project like a technical manual, a report, a book, or a study, then you know how hard it is to keep your writing clearly organized. Somehow, your original structure disappears in a maze of chapter headings, sections, and sub-sections. As your document grows, it becomes even more difficult to maintain a perspective on its changing structure. When the writing job is finally done, the tasks of creating a table of contents, a list of figures and tables, and an index still remain. And if there are revisions, it all must be done again. Take heart — WordIndex II is available to help with these dreary chores.

WordIndex II by Borland International is a word processing tool designed to be used with MicroPro's WordStar word processing program. WordIndex II provides document structure, format control, automatic Table of Contents and Index generation for documents written using WordStar. WordIndex II generates text files that are WordStar compatible and that may be re-edited and/or printed using WordStar.

WordIndex II is a second generation product (as might be guessed from its name) developed in Denmark for writing technical manuals and other formal documents. It includes a 70-page, detailed user's manual and an 8" disk (in standard CP/M-80 distribution) containing the program, overlays, and an example document (the user manual itself). The product has been bundled by several microcomputer manufacturers including Morrow and Molecular Computer, and MicroPro will be offering a limited version called StarIndex as a WordStar menu-linked add-on.

WordIndex II is a text file post-processor. The user places non-printing control commands for WordIndex II in the WordStar source text file to specify format, chapter and section titles, figures, tables, and key words for indexing. After the document is written, it is submitted to WordIndex II, which processes the source file and outputs two new files — a .TOC file containing a generated table of contents, tables, and figures; and a .WI file containing the processed document and an appended subject index.

On a 4MHz Z80 Alspa computer with dual 8" double density drives, the 70-page user manual takes 18 minutes to process (3.8 pages per minute). This is due to the extensive use of key words and variables in the manual. A simple document with only a few new chapter and section headings per page and a few key words will process at a rate of up to 15 pages per minute.

Preparing a WordStar document for use with WordIndex II is quite simple. WordIndex II makes use of the WordStar dot command for inserting comments, which causes WordStar's editing and printing programs to ignore any line beginning with two periods or 'dots' (..). Most WordIndex II commands (or directives) take the form of ...X and are invisible to WordStar. These directives do not take up text lines in the final printed output document.

WordIndex II supports nine levels of document structure, including Chapters, Sections, Sub-sections, and six more levels. It is obvious that this level of organization is far beyond that ordinarily required for commercial writing, although legal contracts and military or government manuals might require the full nine levels. The format of headings for each level is user definable.

Using the program is quite simple — just add the appropriate directives on the line preceding titles, headings, tables, or figures. Key words for indexing are isolated either by using a directive (...K) just prior to the paragraph they appear in or by enclosing them in the text with special print control characters (↑P↑K subject word(s) ↑P↑). Similar commands allow identification and indexing of figures and tables, as well as chapter and section headings.

WordIndex II has several powerful features. The include feature allows several files to be 'called' from a master file and included in a final document, allowing chapters to be broken up into manageable pieces for editing and combined for indexing and output. Or, the user may use separate files and define the starting values of chapter number, page number, and other internal counters for each.

The program also supports text 'variables'. This allows the user to define headings, page and chapter cross-references, and other location- or document-dependent items as WordIndex II variables to be inserted in the text when the final indexed print-out version is generated. This allows easy revision of a cross-referenced text or the use of a standard format for several different texts (by just changing the assigned value of the variables like Title, Chapter, page heading, etc.).

As a fulltime technical writer, I have used WordIndex II on all types of documents from a few pages to 150k files and have not experienced any 'bugs'. I find it valuable to create an outline in WordIndex and then 'fill-it-in' as I write. As I develop drafts of the document, I can see easily what areas need better organization by looking at the structured table of contents (TOC). It can also be helpful when revising an old text to add the WordIndex directives and then review the organization of the document from the generated TOC before it is revised.

WordIndex II has a couple of restrictions in output formatting options which limit the current version's usefulness. Chapter, section, and level numbers are automatically added to each heading as the file is processed by WordIndex. This feature cannot be disabled in the current release. In addition, the automatic numbering of figures and tables is always in the

format CHAPTER-NUMBER (i.e., table 2 in chapter IV would be designated as IV-2). These two fixed formats place some restrictions on the writer's style. Of course the output file can be re-edited to change or eliminate the numbering, but this is an added step. Borland states that the next version of WordIndex II will allow disabling the heading numbering for final printout (it is quite valuable during draft editing) and will allow consecutive (rather than chapter) numbering of figures and tables. Existing copies may be updated for a small fee.

Overall, WordIndex II is a well- documented, useful, and efficient writing tool. The program provides many features which may be used or ignored, allowing both casual and professional writers to benefit from its use. WordIndex II should be of interest to technical writers, lawyers, proposal writers, authors, and others with a need to organize and document their work. The program will be available for IBM-PC compatible 16-bit computers in late 1983.

Product: WordIndex II
Release 0.87 5/82

From: Borland International

Price: $195.00

OS: CP/M-80 2.2 (Avail. for CP/M-86 and MS-DOS in Oct. 1983)

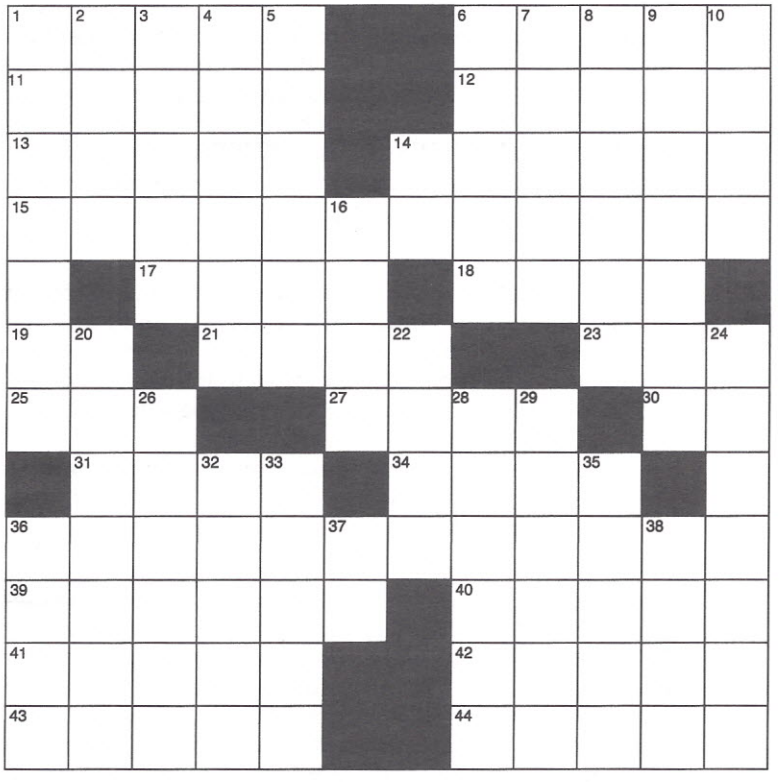
CPU: 8080 (8086-8088 avail in Oct. 1983)

Design: Written in Pascal for easy mobility to other OSs.

Purpose: Word Processing Tool - for use with WordStar

Crossword Puzzle

by Crescent Varrone



Across

1. Technical editor, Al __.
6. Text editor.
11. Journal.
12. Perjurers.
13. Beer __.
14. Chops finely.
15. Regular *Lifelines* author.
17. Roman violinist?
18. Abyssinian weight.
19. Programming Language (abbr.)
21. __ cracker.
23. Land of Cain.
25. Cartographer's product.
27. Suffix (pl.).
30. 3.1416...
31. Mardi __.
34. Labatt's and Bass pale.
36. *Lifelines* author of PL/I articles.
39. They go solo.
40. Device that permits the flow of electricity in one direction only.
41. Another name for ARose?
42. Fitzgerald and Raines.
43. King of Judea.
44. Judges.

Down

1. __ error; __ operating system.
2. Bits of thread.
3. Made of grain.
4. Longs for.
5. Zoological prefix with artia or odon.
6. Subject of 36 Across article in August *Lifelines*.
7. Russian city.
8. Aix-la-Chapelle.
9. Woody plant apex.
10. Estonian Soviet Socialist Republic (abbr.).
14. Musical notation for less than forte.
16. Knot or point of convergence on a graph.
20. New French dance done in dark?
22. Oriental nurse.
24. Wander from the subject.
26. Adam was the first.
28. Avoided.
29. Forgetful, ravaged with age.
32. Combining form for $C_2H_4O_2$.
33. Dried (poetic).
35. Long scarf.
36. Dull, lifeless (slang).
37. Before coll.
38. This cheese is "made" backwards?

Feature

Update On Power

by Robert Kowitt

In the May 1983 issue of *Lifelines*, I reviewed a program published by Computing!, a San Francisco company. My review was enthusiastic for a program I find to be one of my most useful acquisitions. Since getting POWER, I have seldom used PIP or DDT, and I never use STAT.

The magazine publishing business being what it is, there had been several months' delay between writing the article and its publication. In that time Pavel Breder, the author of POWER, did not rest on his laurels. The man is obviously a software hacker who cannot leave a good thing alone. There are many of us who write programs that, once written and working, cannot be run again without making additional modifications. He must be one of those.

I represented one of our local dealers at a computer show at the New York Coliseum last September and was pleasantly surprised to find a booth for POWER. I had the opportunity to talk several times during the four-day show to Clyde Steiner, who ran the booth, and found that he had a great impact on the appearance and operation of the final product.

I wrote this article because I felt that if POWER was good before, it is certainly super now. If you did not read the previous article, get it, because I want to avoid repetition and will try to describe only the changes and additions.

The original small (4x6x1/8") manual has been completely rewritten by Shari Steiner and expanded to 8-1/2x11x1/2", 112 pages with no wasted space. Each command is explained and examples given. The layout is well done and, in addition to a table of contents, there is a functions index and an index of commands. With the functions index, you can look up the command that will accomplish what you want even when you don't know the command itself. The index of commands is just that: a brief description of the command.

All DIRectory displays, for all directory run operations, are now in alphabetical order. This greatly improves operations, as I wished for in the May article. In addition, through the LOG command, this sort can be set for different modes:

(1) By file name
(2) By file name with system files last
(3) By file type
(4) By file type with system files last

The system files are marked with parentheses and READ ONLY files are marked with an asterisk '*' to differentiate them from the other files. To get the directories on all disc user areas, type DIR [u]; for all disc drives on line DIR [x] or for both DIR [ux].

When one is RUNning another program from POWER, POWER now establishes a $$$.SUB file that will return command to POWER upon completion of the called program. The RUN command will accept the same command line syntax as CP/M or, should there be no parameter needed for the command file, RUN, typed alone, yields the directory for selection. Typing its number loads and runs the program.

COPY has been modified. Although I try never to run a program without thoroughly reading the manual, I made an exception this time, since I had been using POWER for several months.

The XUSER command allows you to specify the destination user area as it did in the previous release. While using it, I often thought that there should be a way of moving a file to a different area instead of copying it. It was inconvenient to have two copies on the disc, not to mention having to erase one copy from the old area after the copy, but the move operation should have been much simpler and faster: merely a change in byte 1 of the directory listing. Lo and behold! When I used XUSER the next time to move a file to the same disc, I saw the unexpected:

(C)opy or (M)ove: (Thank you, Pavel)

After a copy has been made, the directories on both discs indicate that a copy has been made and in which direction it was done. The source disc directory indicates the change by displaying a '>' instead of a '.' between the filename and filetype. This indicates that the file was copied 'from'. The destination directory listing is indicated by a '<' as the file copied 'to'. This modification is performed by setting the high order bit on either the first or second character of the file name. This has additional uses through the application of the SET command.

SET is a new command. This powerful command allows you to set the highest bit, bit 7, of any character in the filename or filetype. Some of these are already in use by CP/M to set the read/only or read/write status of the file. For this purpose, these are redundant, but there are other important uses for SET within POWER. I have found it convenient to set the high bit of the eighth character in the file name of a group of dBASE II application files (these have filetypes CMD, OUT, IO, ADD, FAU, and VAL as used by Fox and Geller's Quickcode). The command to do this is:

SET [-8]

The directory of files that DO NOT have this byte set is displayed, and I can select those to set from the numbered directory listing that is the standard format for POWER's operation (no mistyping names). When I want to copy them to a disc for transporting to a client without the other files on the disc, I do not have to select the files I want. I just issue the command:

COPY [+8]

This copies only those files with this byte altered. Should I want to copy only those files that are NOT set:

COPY [-8]

To reset the 8th byte:

SET [+8]

will display the directory of only files that have this byte altered. Any of the commands, REN, ERA, etc., can be used in the above manner. Typing SET alone displays the bit settings of the files you select.

There is another application of the '<' or '>' as placed into the directory through the copy operation. Let us assume we have a virgin disc that has not been copied with POWER. We decide to create more than one identical disc containing some, but not all, of the files on this disc. After copying with POWER, the source files have been marked '>' by the copy operation. To copy the same files again, log in the new disc and

COPY *** [+1] (the *** bypasses the selection query)

Only those files that were marked as having been copied before will be copied.

In the last few months, I have been increasing use of the SPEED command. I have upgraded my system. I was running a Vector Graphics Mindless Terminal with their Flashwriter II memory mapped video board. The display rate, for some reason (I'm not big on hardware), appeared to be about 2400 baud. When I upgraded to an Advanced Digital Super Six, I decided to go with a new terminal also. My Visual 50, at 9600 baud, now scrolls so fast that I cannot comfortably read the screen as I could before. SPEED allows me to adjust the display on the fly from normal 0 to a creepy 9. Typing a new number at any time during display changes the rate, the space bar stops the display and any other key continues it.

For those of you who will be giving POWER to inexperienced computer users, there is a RETOOL utility (not one of the regular commands). RETOOL permits renaming or removing of POWER commands. The ERA function, for example, can be renamed DL. If you don't tell the computer operator, files cannot be erased. Incidentally, don't tell him/her how to EXIT (change the command to QU), and all other functions will always return to POWER. If you install an autoboot, you can protect your system from inadvertent (or otherwise) tampering. In addition, there is a PASSWORD program provided to prevent unauthorized use of some of the commands and to lock unauthorized users away from user areas greater then User 15.

I have been using Ward Christensen's DU to repair and modify directory entries for some time. Now I use POWER for the same function and it is easier to use for the job I have to do. Use the READGR command to load the directory into memory above POWER:

READGR 0 4000 32 (32 sectors)

modify the memory image and then:

WRITEGR 0 4000 32

Be careful with this one. It's easier to destroy a disk with a magnet but not much easier. Should you write the wrong thing to the directory area, your disk is a frisbee **unless** you know how to fix it with POWER.

There is an excellent tutorial on recovering data from damaged discs. POWER's ability to access the discs directly, by group or track, permits recovery from discs that otherwise would be given up for lost and reformatted. This recovery can be made but it's really not one of those things I like to do, given the choice. However, it's nice to be able to do so if it becomes necessary. It's easier to be super careful in the first place.

The only complaint I have about READGR and WRITEGR is one that may be explained by other requirements of the software. Since CP/M only uses the disc in whole groups, why do we have to specify the number of sectors? Could we not READGR 0 4000 2 and have two whole groups read into memory?

An example:

One of the requirements of the Advanced Digital System is that the CPM.SYS that contains the operating system be the FIRST file on the disc in order to warm boot on Control-C. That is, CPM.SYS must start on Group 2 (Groups 0 and 1 contain the disc directory). I had some discs with other files on them before the system was copied to them. I used POWER. I read Groups 0 and 1 into memory at 4000H. The first directory entry looked like this:

```
4000: 00505F57 45522020 20415254 00000075 .POWER ART...u
4100: 02000300 04000500 06000700 08000900
```

I had to COPY POWER.ART to another area of the disc which changed the first byte from 00 to E5. Now I had to find the CPM.SYS on the disc and since it was at:

Groups BE,BF,C0,C1,C2	(5 groups)
READGR BE 4000 80	(80 sectors in five groups, since they were contiguous)
WRITEGR 02 4000 80	

This effectively moved the CPM.SYS into the first position on the disc and was faster than copying the entire disc. I took more time to write the description of the operation than the actual doing.

Wish list

When RUNning a COM file thru POWER, you are asked to enter a number from the directory. Why not ask for another number for a parameter to the command file to be run? This new file will be added to the CP/M command line. Thus, if 18 is WS (WordStar) and 6 is POWER.ART (this article on my disc), I would, when POWER responds to my RUN with:

Select?: (I would enter) 18 (cr) 6 (cr) and the command line inserted by POWER would be:

WS POWER.ART

Typing 18 (cr) (cr) could indicate no parameter needed.

Another idea: allow the manual entry of a parameter from keyboard after selecting the RUN directory number.

I still would like to see the TESTing of a disc bypass the remaining sectors of a group when a bad sector has been found. This would speed the disc test considerably.

MENUPOWER

MENUPOWER is a subset of POWER and is sold at a slightly lower price for those users with a limited need. It also uses commands whose names are easier for the new user to understand.

The commands supplied:

DIR	— same as POWER.
ERASE	— same as POWER.
RUN	— same as POWER.
TYPE	— same as POWER.
COPY	— same as POWER.
RENAME	— same as POWER.
TEST	— same as POWER — tests a disc for bad sectors.
CPM	— same as POWER's EXIT — to CP/M.
RESTORE	— same as POWER's RECLAIM — restores erased files.
COMPARE	— same as POWER's CHECK — do a checksum.
PROTECT	— same as POWER's SETRO — set Read Only status.
UNPROTECT	— same as POWER's SETWR — set Read/Write status.
SPACE	— same as POWER's STAT — give amount of space on disc.

MENUPOWER has the password protection feature and, in addition, does some things in a more direct way than either CP/M or POWER. An example of this is moving to another disc and another user area. Under CP/M, you must type, for example, C: (cr) USER 4 (cr) to go to disc drive C, User 4. Under MENUPOWER it is: C4 (cr) and you're there.

Copying is simpler also. When the program asks the destination drive, you may enter: C4, and the copy will move to the other disc and user area without requiring POWER's XUSER command.

What MENUPOWER does, it does well, as a true child of POWER, but it is not intended for the sophisticated computer user. However, it is a subset of POWER and not a separate utility. For the small amount saved, about $20, I would recommend that everyone get the more complete POWER. Incidentally, multiple use licenses are available to those offices that use more than one computer. The dealer or installer can set up POWER on his customer's computer and modify the accesses to commands through the supplied RETOOL utility, reserving the powerful and potentially damaging commands to the selected few. With this done, the resulting POWER will do what MENUPOWER does without sacrificing those powerful tools that may be needed by a programmer in an emergency.

Computing! will soon be releasing two more programs: DOCUPOWER and SALES/POWER. If they are written with the thoroughness and proficiency of their current releases, we can expect big things from Computing!. I am looking forward to the opportunity of letting you know about them in the near future.

POWER and MENUPOWER are available from:

Computing!
2519 Greenwich Street
San Francisco, CA 94123

(continued from page 2)

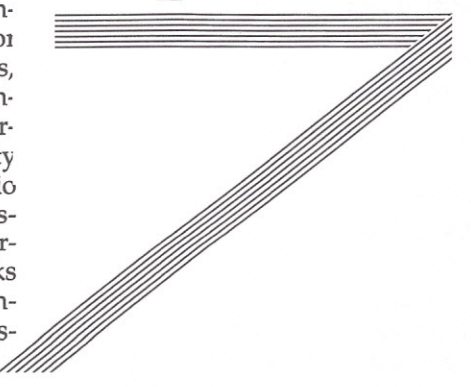
that QST, 73, et al. are seldom visible, or review the diminishing list of ham radio equipment manufacturers, to realize that there is considerable cause for concern. Store shelves are straining under the ever increasing number of computer magazines and books, none of which have any direct correlation with ham interests or activities.

Instead of engaging in endless debates about what is happening one would be well advised to recognize that this important national asset is in serious danger. Furthermore there are active efforts afoot to lower the entrance requirements which, at first blush, would appear to be a logical step. However, while this might mean a larger membership it would undoubtedly result in a loss of quality. The point to be made is not what the ultimate viability of ham radio may be, but rather to encourage positive developments such as additional emphasis on microcomputer uses in the ham bands. One has only to note that significant numbers of potential ham candidates are steadily being lured away from ham radio and gravitating towards the microcomputer to conclude that the merging of these two technologies is an obvious solution.

Packet radio offers a number of exciting possibilities for software transmission, for example, and micros lend themselves to a wide variety of signal processing, conditioning, etc. which are applications of great interest in radio communications. As to the use of two meters, listen in on the local repeaters and draw your own conclusions. This column will continue to champion the cause of microcomputers and all classes of communication as important realms for amateurs as well as professionals, and its author will continue to campaign for the FCC to broaden the horizons to afford the best opportunity for microcomputers and ham radio to insure that important national assets such as skilled ham radio operators, extensive repeater networks and microcomputer technology enjoy a mutally beneficial and synergistic future.

The rumor mill continues to grind out alleged facts about the IBM Peanut. In part, the Peanut is said to have a single 5¼" drive, a new release of MSDOS, add-on Winchester, optional joysticks, max of 128K RAM, full keyboard (if you are willing to punch enough keys), wireless keyboard (uses infrared), 8088 processor, no socket for the 8087 (may be available on optional card), 18 connectors, etc.

Next time we will review in detail the IBM Peanut but for now in the immortal words of Mr. Leonard, "Later!".



Feature

Learning Not To Swear At Your Delete Key

Part 2

by Mark Gardner

You may remember my attack on the DELETE key in the July 1983 issue of this magazine. I have now patched my disk, as I promised, so I never have to run DELPATCH again. However, that is not why I'm appearing with another article of the same name: this time I want to show you how you can provide the same DELETE key fix for MBASIC.

The previous article was very short, just because the solution for MBASIC was not easily forthcoming, and I couldn't work it out in time for inclusion. In Part One, the BDOS itself was patched, to cause the system to treat the DELETE key just like the BACKSPACE key, i.e., a true backspace and blank, rather than the atavistic echo typical of the DELETE key under CP/M. That fix only works when the BDOS is called for line editing. MBASIC calls for characters one at a time and processes them itself, hence a separate solution is needed.

Part of my delay in treating the MBASIC problem had to do with personal schedule, but largely it was the hammer syndrome ("To a man with a hammer, all problems look like nails"). I was looking for a single location in the MBASIC.COM file that I could patch. I don't think there is one. There are several dozen occurrences of compares for values that look like DELETE and BACKSPACE keys, and mostly my experimentation with them caused either no change, or system catastrophe.

Consequently, I put my hammer away, and reconsidered the problem from another viewpoint — that of intercepting MBASIC's calls to the BIOS in order to allow replacement of the DELETE key with the value for the BACKSPACE key. This is not a good solution. I avoided it with Part One because the DELETE key disappears. If you need it, you don't got it. So, in the program provided with this article, the patches are removed when a SYSTEM command is used to exit from MBASIC. That way, pro-

```
; Program to patch bios on entry to MBASIC, restore on exit.
; Makes DELETE Key function correctly (like backspace key)
; but restores, since other programs (WordStar) need both.

BDOSBS  EQU   0D000H              ; BASE OF BDOS
BIOSBS  EQU   0DE00H              ; BASE OF BIOS
PATCBS  EQU   BDOSBS-180H         ; LOCATION FOR PATCH ASSEMBLY
BDOS    EQU   5                   ; SYSTEM BDOS CALL LOCATION
TRANSI  EQU   100H                ; TRANSIENT PROGRAM AREA
PRNSTR  EQU   9                   ; BDOS STRING OUTPUT
OPEN    EQU   15                  ; BDOS FILE OPEN
READ    EQU   20                  ; BDOS FILE READ
SETDMA  EQU   26                  ; BDOS SET DMA

        ORG   100H

        LXI   D,IDSTRG            ; PRINT PROGRAM I.D. STRING
        MVI   C,PRNSTR
        CALL  BDOS

        LXI   H,BDOSRE+OFFSET     ; SET UP BDOS CALL TRAP
        SHLD  BDOS+1
        LHLD  BIOSBS+4            ; SAVE OLD WARM BOOT ADDRESS
        SHLD  WRMSAV              ; (WITHOUT OFFSET, NOT MOVED YET)
        LXI   H,WARM+OFFSET       ; AND STORE NEW ONE
        SHLD  BIOSBS+4

        LHLD  BIOSBS+10           ; SAVE OLD CONIN ADDRESS
        SHLD  CINSAV
        LXI   H,CIN+OFFSET
        SHLD  BIOSBS+10           ; AND STORE NEW ONE

        LXI   H,MOVBAS            ; MOVE THE LOADER AND PATCH
        LXI   D,MOVLOC            ; TO THE UPPER MEMORY AREA
        LXI   B,MOVSIZ
MOVLUP:
        MOV   A,M                 ; GET A BYTE OF PATCH
        STAX  D                   ; AND MOVE TO PATCH AREA
        DCX   B                   ; SEE IF DONE
        MOV   A,B
        ORA   C
        JZ    LOAD + OFFSET
        INX   H
        INX   D
        JMP   MOVLUP

;       ORG   PATCBS-(BDOSRE-MOVBAS)          ; TEMPORARY SPACE
MOVBAS:
MBAFCB:                           ; FILE CONTROL BLOCK FOR MBASIC
        DB    0,'MBASIC COM'
        DB    0,0,0,0,0,0,0,0
        DB    0,0,0,0,0,0,0,0
        DB    0,0,0,0,0,0,0,0

        DB    'PATCHING COMPLETE — GETTING MBASIC',13,10,'$'
        DB    'GOING TO MBASIC',13,10
        DB    'PATCHES ARE UNDONE WITH A "SYSTEM" EXIT',13,10,'$'
        DB    'MBASIC BIOS PATCHER V1.0 MRG 8/83',13,10,'$'
```

►

grams like WordStar, which intelligently use both keys in specific ways, have them available without the need for a cold start.

Where the patch goes

Sorry, no simple DDT procedures this time. There's just too much to do to be even close to convenient with a manual approach. The program provided installs the patches, and leaves in memory the intercepter code that changes DELETEs to BACKSPACEs, and that catches the warm boot when MBASIC is done, so the patches can be removed.

The program loads and executes at location 100H in the TPA, not surprising. The BDOS call address at locations 6 and 7 is changed to point to the BDOSRE entry in the program; the BIOS jump table entry for warm boot is changed to point to the WARM entry; and the BIOS jump table entry for console in is changed to point to the CIN entry.

[Rules! Rules! Right there in the CP/M manual are admonitions to access the BIOS only by calls to the BDOS, or your code might not be upward compatible. MBASIC does *not* access 'console in' by calls to location 5. What a surprise. That's why my program needs two traps, one for access via the BDOS, and one for access via the BIOS. I suppose I could have taken out the first, but I'm not absolutely sure that MBASIC never makes such a request via the BDOS, so I left it in.]

After the patches are made, a large portion of the program is moved into the upper part of the CCP. A portion of this code loads MBASIC into the TPA, and transfers control to it. Another portion consists of the three trap routines (WARM, BDOSRE, and CIN). These last become protected against overwriting by MBASIC, since their start address is written at locations 6 and 7 in the initial patching. The approach taken allows MBASIC to reuse all memory except that absolutely needed for the duration.

The MBAS program

The program is called MBAS for obvious reasons, and after it's assembled and loaded you can use it exactly like MBASIC. The command line

```
LOAD:                                   ; LOAD MBASIC AND TRANSFER
                                        ; CONTROL
        LXI     D,STRTLD + OFFSET
        MVI     C,PRNSTR
        CALL    BDOS
        LXI     D,MBAFCB + OFFSET       ; OPEN THE FILE
        MVI     C,OPEN
        CALL    BDOS
        LXI     D,TRANSI                ; LOAD INTO THE USER TPA
LOADLP:         ; LOAD LOOP
        PUSH    D                       ; SAVE FOR INCREMENTING
        MVI     C,SETDMA
        CALL    BDOS
        LXI     D,MBAFCB + OFFSET
        MVI     C,READ
        CALL    BDOS
        POP     D
        ANA     A                       ; SEE IF END OF FILE
        JNZ     NOMORE + OFFSET         ; IF SO, TRANSFER CONTROL
        LXI     H,128                   ; POINT TO NEXT SECTOR AREA
        DAD     D
        XCHG
        JMP     LOADLP + OFFSET
NOMORE:                                 ; END OF FILE HAS BEEN REACHED
        LXI     D,DONELD + OFFSET
        MVI     C,PRNSTR
        CALL    BDOS
        JMP     TRANSI

;       ORG     PATCBS                  ; BEGINNING OF PROTECTED SPACE
BDOSRE:         ; TRAP ROUTINE FOR CALLS TO BDOS
        MOV     A,C                     ; SEE IF CONIN OR RAW I/O
        CPI     1
        JZ      ITSCON + OFFSET
        CPI     6
        JNZ     BDOSBS + 6              ; IF NEITHER, NO MORE
                                        ; INTERFERENCE
        MOV     A,E
        INR     A
        JNZ     BDOSBS + 6              ; (ON RAW I/O, E = FF MEANS INPUT)
ITSCON:         ; IT IS CONSOLE INPUT
        CALL    BDOSBS + 6              ; EXECUTE BDOS CALL, BUT
        CPI     7FH                     ; IF RETURNED KEY IS DELETE KEY
        RNZ
        MVI     A,8                     ; THEN REPLACE WITH
                                        : BACKSPACE KEY
        RET

WRMSAV: DW      0                       ; LOCATION TO SAVE BIOS WARM JUMP
CINSAV: DW      0                       ; LOCATION TO SAVE BIOS
                                        : CONIN JUMP

WARM:           ; TRAP ROUTINE FOR BIOS WARM START
                ; UNDOES ALL PATCHES BEFORE RETURN TO SYSTEM
        PUSH    H
        PUSH    D
        PUSH    B
        PUSH    PSW
        LXI     H,BDOSBS + 6
        SHLD    BDOS + 1
        LHLD    WRMSAV + OFFSET
        SHLD    BIOSBS + 4
        LHLD    CINSAV + OFFSET
        SHLD    BIOSBS + 10
        POP     PSW
        POP     B
        POP     D
        POP     H
        JMP     BIOSBS + 3
```

MBAS alone will bring up MBASIC in the direct mode, and the command line MBAS filename will bring up MBASIC with filename running. In either case the patches are installed and the trap routines are in place.

The program is designed so that you only have to provide two numbers that are particular to your system. These are the values for BDOSBS and BIOSBS shown as the first two equate statements in the listing (Figure 1). The BDOS base can be found by the procedure given in the July article. It's likely the BIOS base is just +0E00H above, but you can check with DDT. The beginning of the BIOS is very distinctive, being some 17 jump instructions in a row.

The third equate defines how far below the BDOS the trap routines will start. This is the only recursive number in the program. Near the end of the listing, you'll see an equate for MOVSIZ. The number in the equate for PATCBS should be close to this. The remaining equates at the top of the program merely define the system call values.

The code from 108H to 123H installs the patches as described above, then the code from 126H to 139H moves the remainder of the code into the CCP. (The CCP is not being patched — it's being overwritten. It won't be needed again until MBASIC returns control to the system, and the warm boot will restore the portions this program destroys.) The locations and quantity to move are all calculated as parameters within the program, so that as changes to the code changed its length, it was not necessary to be constantly recalculating where and how many. This is a nice feature, part of the total approach to getting a piece of operating code into high memory without having a huge .COM file.

The other half of the solution is the value shown as OFFSET. When added to any address reference in the program, the result is the address *after* the code is moved. This is all automatic (by means of the equate statements). It's not as nice as a relocating loader, but it beats 62K .COM files. In fact, it's a fairly clean solution to the problem, and I hope that just this bit of work in the program will be of use to you.

```
CIN:                ; TRAP FOR DIRECT ACCESSES TO BIOS
                    ; CONIN VIA JUMP TABLE
        PUSH    PSW
        PUSH    B
        PUSH    D
        PUSH    H
        LHLD    CINSAV+OFFSET
        SHLD    CINCAL+OFFSET+1
        POP     H
        POP     D
        POP     B
        POP     PSW
CINCAL:
        CALL    0
        CPI     7FH
        RNZ
        MVI     A,8
        RET
MOVLIM:
MOVSIZ  EQU     MOVLIM-MOVBAS
MOVLOC  EQU     PATCBS-(BDOSRE-MOVBAS)
OFFSET  EQU     PATCBS-BDOSRE
        END
```

The moved code contains the MBASIC loader (it has to be well out of the way, since MBASIC loads at 100H) and the traps. The loader merely opens the MBASIC.COM file and repeatedly increases the DMA and reads until an end of file is encountered. At that time, control is passed to location 100H, and MBASIC takes over. If there was a tail in the command line, MBASIC treats it correctly, as if it were invoked directly. This is because the command line tail (at location 80H) is not touched by the patching program.

The moved code also contains the traps. BDOSRE is entered (from location 5) for all calls to the BDOS. If the call is not a console in, either by polled or by raw I/O, then it is passed along without further intervention (the JMP out of BDOSRE will cause the BDOS return to take control back to the calling program). If it is a console input, the BDOS is invoked with a CALL instead, and control returns so that any occurrence of the DELETE key can be replaced with the value for the BACKSPACE key. The BIOS trap, called CIN, always calls the BIOS routine, and control is returned for the same check as in BDOSRE.

The warm boot trap is entered from the BIOS jump table, typically when the SYSTEM command is executed by MBASIC. The patches that were installed are all returned to their original values, and control is passed on to the warm boot routine, which restores the CCP and gives control to it. At this point, all of the MBAS program has been effaced from memory.

What it does for you

You will recall how delighted I was with DELPATCH, that I would no longer have clobbered command lines from the reverse-order echoed characters that appear when the DELETE key is used. The relief inside MBASIC is double, because not only does MBASIC reverse echo the characters, it sets them off in slashes. This is double jeopardy, believe me. It's still possible to get in trouble with the EDIT function, since the D key is now the delete function, and it's only possible to delete in a forward direction along the line, and the slashes will still be around. Even without the patching, any backspacing or delete keying can cause confusion to you in what is actually on the line. When this happens, the best bet is to use a couple of 'L' commands to see exactly what you've got.

No more

Now that my two commonly used tools have the DELETE key fixed, I suppose I'll be content for a while. (WordStar was already okay, or I'd be attacking that.) I hope these exercises prove useful to you. They certainly take away some of the frustration from losing useful programming/debug time, and provide for more enjoyable, less surprising sessions at the keyboard.

Feature

Tips To Follow With T/Maker III

by Suzanne Osterlund

Note: Lifeboat Associates of New York was T/Maker's exclusive distributor up until 1982. T/Maker owes much of its success to this distributor. Now, as a major distributor of T/Maker III, Lifeboat has continued to sell thousands of copies. Because of this close union, a T/Maker monthly column begins with this issue. This column will include news, tips, and other informative bits about T/Maker III integrated software.

In this first column about T/Maker applications, I thought I would give a little background information about T/Maker III and provide two tips for users.

Over a period of three years, Peter Roizen, author of T/Maker, has revised his program to produce what is now known as T/Maker III integrated software. With versions for MS-DOS, PC-DOS, CP/M, and CP/M-86, T/Maker III covers more operating systems and computers than any other integrated package. Users of T/Maker II can easily upgrade to T/Maker III by contacting Lifeboat at (212) 860-0300.

This versatile package provides the user with word processing, electronic spreadsheet, list processing, bar-charting, data transfer, and file management. The user is given a blank screen on which to create (using the full-screen editor) anything he wants. Every function and command can work on any file. Because of this convenience, the user can do invoices, letters, cash flows, tax accounting, proposals, or anything that combines text, numerical analysis or bar charts — all in the same file, eliminating the need to switch programs or files.

The T/Maker program is both powerful and flexible — as such, users have found "tricks" to make the use of T/Maker even better. Following are two tips — the first is designed to enable users to save time when printing lots of letters; the second, to make lines of text more clear and legible by aligning periods in their proper place.

Printing Several Documents

Several businesses stick to one certain method of printing letters. They write a number of letters, save them as individual files, and then print them all out. Here is a new time-saving method for printing a number of letters using one file called "print-all" which contains the following:

```
PRINT NONSTOP JONES DO
PRINT NONSTOP TAXLETR1 DO
PRINT NONSTOP TAXLETR2 DO
PRINT NONSTOP SIMPSON DO
PRINT NONSTOP SIMPSON DO
PRINT NONSTOP MOTHER
```

First, SAVE this file, then type DO, and it prints the letters Jones, Taxletr1, Taxletr2, two copies of Simpson, and Mother. Extra copies of any of the letters are made simply by duplicating the print command line for that letter once for each extra copy desired.

The next time one needs to print out some letters, this same file can be used — only the names need to be changed. If the paper is hand-fed to the printer, one can simply leave out the "nonstop" and then the familiar printer prompt will appear before each letter.

<<< >>

Tricking Align Into Leaving Periods Alone

Those users who have been around since the first version of T/Maker can remember the days when periods were aligned with only one space following them. Now, two spaces follow a period, but, unfortunately, having two spaces isn't always what one wants either. For example, two spaces between "Mr." and "Jones", or "P." and "O." would look odd. Well, for the users who have created a print.utl file from the instructions in the back of the manual, here is a simple solution:

Between the period and beginning of the next word or letter, do not use a plain space. Instead, use the high-bit key, followed by a tap on the space bar. This keystroke combination creates something that looks and prints like a space, but is seen by T/Maker as a character. Therefore, it does not change the spacing. Not only does this method serve nicely for keeping periods closer together, it can also be used to keep two words from being separated by ALIGN. One must remember, however, that as far as T/Maker is concerned, any words joined by these high-bit spaces are looked upon as single words, for better or worse.

If you are a T/Maker user who has tips to share, please write to: "T/Maker Tips," c/o Lifelines, *or write to T/Maker Company at 2115 Landings Drive, Mountain View, California 94030. T/Maker also publishes a bimonthly user's group newsletter packed with helpful information on practical applications.*

A Call For Manuscripts

Perhaps you've done some writing before. Or maybe you've always wanted to write. It could be that reading *Lifelines/The Softward Magazine* has given you some ideas on what you have to contribute. We're interested in hearing what you have learned, and so are other readers. Whatever *serious* CP/M-80 compatible software you've been using, we'd like you to write for us. We like to publish both long essays and those short gems which can hold so much important information.

Send us a brief resume of your software experience, and samples of your previous writing, if you have any. (Don't be shy if you're *not* an experienced writer.) Then we can talk about your work and about payment for your efforts. Write or call: Editorial Dept., Lifelines Publishing Corp., 1651 Third Ave., New York, NY 10028. Telephone: (212) 722-1700.

Product Status Report

New Products

PROFORMA

Management Control Concepts
124 St. Mary's
Boston, MA 02215
(617) 267-3592

PROFORMA is a "load and run" forecasting/projection program. It is designed for novices. You are allowed to enter data monthly, quarterly, by constant percentage, by varying percentage, by actual number or in several complexity modes. PROFORMA uses more than 700 pre-programmed conditional commands. It calculates payments, principal, interest on balance due on 120 categories of short and long-term debt and separates each debt into its long and short term element. It allows the user to select straight line, declining balance or sum-of-the-year's-digits depreciation, in addition to accelerated cost recovery. In PROFORMA all financial functions are built in. PROFORMA will compute the amount of tax loss carry-forward, calculate periodic or monthly installment payments, interest expense and loan balance, invest excess cash and calculate interest income, evaluate depreciation by a choice of several methods, and allow input as a percent of sales or estimated value.

Requirements: CP/M or Apple II or III
Price: $395

DISK FIX-DISK EDITOR AND RECOVERY UTILITY

The Software Store
706 Chippewa Square
Marquette, MI 49855
(906) 228-7622

This program can be used to reconstruct files with bad sectors, to recover files from disks with damaged directories, to restore erased files and to do general disk editing. It is menu driven. DISK FIX automatically configures to floppy and hard disks. It can display, edit or copy any sector of a CP/M 2.x floppy or hard disk. Directory on screen editing is provided in both HEXidecimal and controlled ASCII.

Requirements: 8080, 8085, Z80, CP/M
Price: $150

MEDCOS

Business Management Systems
Grand View Road
Sugar Hill, NH 03585
(603) 823-5578

This program monitors patient medical condition and treatment and includes an extensive communications and recall system. The system is designed to manage medical accounts receivable, insurance form preparation and appointment scheduling for a medical practice of up to 99 doctors. MEDCOS is a transaction-based system that features on-demand bills, insurance forms and other reports as well as family billing. All information is also available through on line screen display. The open time accounts receivable system can accommodate up to 32,000 patients, depending on available disk storage. The system uses an ISAM file structure for fast access.

Requirements: 2,000,000 bytes of disk storage min. 5,000,000 rec. CP/M, MP/M, TURBODOS, and MMMOST
Price: N/A

EDSI's FUND ACCOUNTING LEDGER

(404) 955-3374

EDSI's FUNDS ACCOUNTING LEDGER is an accounting system that provides full financial reporting for nonprofit organizations. A complete audit trail is provided for organizations subject to government audit and/or public scrutiny. In addition, separate journals, ledgers, and revenue/expense statements are generated for up to 99 independent funds. A consolidated revenue/expense statement is produced automatically. At a simpler level, the funds' accounting system can be used to track expenses against revenues or budget appropriations. It can serve up to 45,000 accounts per organization.

Requirements: CP/M, MP/M, MS-DOS Price: $750

New Books

HIGH-TECH CONSULTING: A Guide To Making Money As A Computer Consultant

by John Zarrella

Microcomputer Applications
827 Missouri Street
Fairfield, CA 94533
(707) 422-1465

HIGH-TECH CONSULTING provides essential information for anyone considering a career as a computer consultant. It also discusses finding work, setting up a consulting business, the consulting lifestyle, interfacing with clients, setting rates, and collecting payments.

Price: $18.95

CONSULTANT'S LOG

Microcomputer Applications
827 Missouri Street
Fairfield, CA 94533
(707) 422-1465

This book is a monthly recordkeeping logbook. Each log contains space to record mileage, time spent on each consulting project, and entertainment expenses for a complete month.

Price: $3.00

USING 1-2-3

by Geoffrey LeBlond and Douglas Cobb

Que Corporation
7960 Castleway Drive
Indianapolis, IN 46250
(317) 842-7162

This book is designed for users of 1-2-3. It provides an overview of 1-2-3 as well as detailed explanations and examples of 1-2-3's many features, which include keyboard macros, the data base function, and graphics capability. Other topics covered in *Using 1-2-3* are the Data Table command, the Print Graph utility, and the implementation of special-function keys on the IBM Personal Computer.

Price: $14.95

USER'S GUIDE TO MICRO-COMPUTER BUZZWORDS
by David H. Dasenbrock

Howard W. Sams & Co. Inc.
4300 W. 62nd Street
Indianapolis, IN 46268
(317) 298-5400

This book is written for people who don't necessarily care what happens inside a microcomputer, but who want to be able to communicate with people who do. This information-packed guide gives the origins and meanings of hundreds of computer terms. From "abort" to "Z-80," the author covers computer vocabulary in clear, concise language.

Price: $9.95

YOUR FIRST BUSINESS COMPUTER
by Peter Luedtke & Rainer Luedtke

Digital Equipment Corp.
12-A Esquire Road
Billerica, MA 01862
1-800-343-8321

This book provides a basic understanding of computer capabilities, and helps you evaluate your business goals and requirements. It prepares you to explore the marketplace, select a system, and integrate it into your business operations and examines the specific benefits a computer can provide, with a look at the limitations as well.

Price: $15

WORDSTAR IN EVERYDAY ENGLISH
by Maria H. Goudiss

Devin-Adair Publishers
143 Sound Beach Avenue
Old Greenwich, CT 06870

This book is organized by the job that needs doing, not by computer function. It is designed for novices. There are no practice letters or exercises in WORDSTAR IN EVERYDAY ENGLISH. Practice is done on the job at hand and when the first steps are mastered, a usable piece of work will be the result.

Price: $9.95

MICROCOMPUTERS ON THE FARM
by Jack O. Beasley

Howard W. Sams & Co. Inc.
(for address see above)

This book is a hardware and software buying guide for farm businesses. It describes how microcomputers can be used in farm operations for such activities as financial analysis, accounting, recordkeeping, structured decision making, and more. It discusses farm software vendors, dial-up information services and more.

Price: $14.95

New Versions

Nevada FORTRAN v. 3.0

Ellis Computing Inc.
3917 Noriega Street
San Francisco, CA 94122
(415) 753-0186

The compiler, based upon the ANSI-66 standards (FORTRAN IV), contains several advanced features: CHAINing with named and blank COMMON, IF . . . THEN . . . ELSE constructs, TRACE-style debugging, PEEK, POKE, 150 verbal error messages, Arrays of up to seven dimensions and Random Access file support.

Object module loading and chaining takes place in seconds, using the same fast loader as the firm's Nevada COBOL system. The high performance FORTRAN compiler generates 8080 machine language on all CP/M-based systems with at least 48K RAM.

Price: $29.95 plus shipping.

GRAPHWRITER™ v. 3.1

Graphic Communications, Inc.
200 Fifth Avenue
Waltham, Massachusetts 02254
(617) 890-8778

The new version of GRAPHWRITER provides support for both the IBM Personal Computer and the IBM-XT. Support for three new plotters is also included — the X/Y749 from IBM, the Calcomp 84 from Calcomp, and the Sweet-p from Enter Computer. The package already provides support for the Hewlett-Packard 7470A and the HP7220C plotters as well as the IBM X/Y750. Performance speed on both the IBM-PC and the XT has also been enhanced.

Version 3.1 has added features that will increase the graphic capability of a personal computer user.

Version 3.1 will be sent to all registered GRAPHWRITER users without charge. Future GRAPHWRITER upgrades will include additional plotter and printer support, direct interface with the newly-announced Polaroid Palette, an instant slide-making device, and additional performance enhancements.

Price: N/A

More New Versions

1. CP/M Workshop . . . . . v. 2.0
2. Lattice "C" Compiler (MS-DOS only) . . . . . v. 1.04a now includes PLIB-86.EXE
3. HALO (for IBM-PCDOS) . . v 1.33
4. UNIVAIR Insurance Agency System w/PAYABLES . . . . . v. 2.02
5. COBOL-80 . . . . . v. 4.66
6. ASCOM-80 . . . . . v. 2.26
7. PLANFIN . . . . . v. 3.12
8. PROFIN . . . . . v. 3.12
9. FORMULA II . . . . . v. 2.2

Change of Address

Please notify us immediately if you move. Use the form below. In the section marked "Old Address," affix your *Lifelines* mailing label—or write out your old address exactly as it appears on the label. This will help the *Lifelines* Circulation Department to expedite your request.

New Address:

NAME

TITLE

COMPANY

STREET ADDRESS

CITY STATE

ZIP CODE

Old Address:

NAME

TITLE

COMPANY

STREET ADDRESS

CITY STATE

ZIP CODE

Users Group Corner

Editors Note: We hope you will write in and give us information about your users group or computer club. Our Users Group Corner is designed to help you find computer clubs in your area or new clubs that your existing club can exchange information with.

Dayton Microcomputer Association, Inc.
c/o Dayton Museum of Natural History
2629 Ridge Avenue
Dayton, OH 45414-5499

DMA meets at 7:30 p.m. on the last Tuesday of each month (except December) at the Dayton Museum of Natural History. Annual dues for renewing members are $12 (collected in June). New members pay $1 for each month remaining until June 30. Dues include a subscription to the Data Bus.

Micromint MPX-16 Users Group
c/o Jouka Nuora
Reitmaa Engineering
Kumpulantie 1
42100 Jamsa
Finland

We want to get together and form a users group which distributes solutions to MPX-16's problems and puts some pressure on the manufacturer.

Please write us and give your complete name and address, a list of your hardware and software (including version and serial numbers), problems you have encountered, and solutions and patches, if any, you have been able to make. Include two dollars (use registered mail) for our expenses for copying and mailing. You will then receive: 1) a list of users and addresses located nearby; and 2) summary of all problems and solutions that can be found out in this way.

CPro Users' Group
P.O. Box 1474
Woodbridge, VA 22193

The primary purpose of the group is to share technical information, solutions, and special purpose applications.

In addition to a monthly newsletter, both a bulletin board system (with RCP/M and member list publication are planned. The practice of reinventing the wheel will be under direct attack, with members sharing their solutions and discoveries.

Optional: Charter memberships are available for only $10.

G.W. Alpert
Microcomputer Owners of Australia
P.O. Box 2489v G.P.O.
Melbourne, Victoria, 3001
Australia

We are one of Australia's fastest growing computer groups. We are attempting to make available to our members as large a user group bank of software as possible. We would like to swap software with any user group. If you send us any software you have, we will send you all the software we have in a particular area (e.g., Apple, TRS-80, etc.).

Please note that diskettes should be posted securely, customs declarations should be marked "sample only—no commercial value," and items should be sent airmail.

We look forward to receiving your user group software and exchanging ours with you.

CPMUG
1651 Third Avenue
New York, NY 10028

The complete CPMUG™ catalog, which consists of nearly 100 volumes, is available for $10, prepaid to the United States, Canada, and Mexico; $15, prepaid to all other countries. (All checks must be in U.S. currency drawn on a U.S. bank.)

The following description is of Volume 84.

MODEM version 7.65
XMODEM version 5.0

NUMBER	SIZE	NAME
		CATALOG.084
CONTENTS OF CP/M VOL.084		
		ABSTRACT.084
Abstracts of files on this disk		
		CRCK.COM
Program for checking CRCs of files		
		CRCKLIST.084
CRCKs of files on this disk		
84.1	16K	MODEM7.DOC
Documentation for MODEM program (from CP/M UG Vol. 79)		
84.2	14K	MODEM76.LIB
Macro library used with MODEM 7.65		
84.3	6K	MODEM76.SET
Instructions for "hot-patching" MODEM		
84.4	63K	MODEM765.ASM
Macro assembler source code for MODEM 7.65		
84.5	10K	MODEM765.COM
Object code of MODEM 7.65		
84.6	18K	SEQIO22.LIB
Macro library used with XMODEM 5.0		
84.7	3K	XMODEM47.DOC
Documentation for XMODEM program		
84.8	49K	XMODEM50.ASM
Assembler source code for XMODEM 5.0		
84.9	1K	XMODEM51.FIX
Notes on a bug fix for XMODEM		

Note: MODEM program requires assembly with Digital Research MAC macro-assembler. XMODEM may be assembled with ASM.COM if the "logging" feature is disabled, otherwise MAC is required for assembly.

Software in the library, obtainable exclusively on diskettes, is available for a prepaid media and handling charge, as follows:

FORMAT	DESTINATION
8" IBM	U.S., Canada, Mexico—$13
8" IBM	All other destinations—$17
North Star/Apple	U.S., Canada, Mexico—$18
North Star/Apple	All other destinations—$21
KAYPRO II	Same as North Star/Apple

PLEASE CLEARLY SPECIFY THE FORMAT YOU WANT WITH YOUR ORDER

This payment covers the cost of the diskette(s), packaging, and postage. Domestic shipping is via UPS where a full street address is given; all other orders are via U.S. Postal Service.

Answer To Last Month's Puzzle

1 L	2 I	3 F	4 E	5 B	6 O	7 A	T
8 O	R	A	T	O	R	Y	■
9 O	M	I	C	R	O	N	■
10 P	A	L	E	S	T	R	11 A
■	■	12 E	T	■	13 U	A	R
14 N	15 O	D	E	■	16 N	N	E
17 E	D	■	18 R	19 U	D	D	■
20 W	O	21 M	A	N	■	■	22 G
23 I	R	A	■	24 C	25 R	26 C	L
27 N	L	■	■	28 T	H	A	I
29 P	E	30 L	31 E	■	32 O	C	T
33 U	S	E	R	S	■	34 D	C
35 T	S	A	R	■	36 A	S	H

LIFEBOAT™ Associates:
The full support software source.

Reach for the programming horizon of the 80's with Lattice™ C, the fastest C compiler.

Set the course of your next software project towards greater power and portability by selecting Lattice C, recognized as the finest and fastest 16-bit C compiler. Lattice C, the full implementation of Kernighan and Ritchie, continues to lead the way by announcing full memory management up to 1 Megabyte. Major software houses including Microsoft, MicroPro and Sorcim are using Lattice C to develop their programs.

Lattice C is available for a wide variety of 16-bit personal computers including IBM®, NCR®, Texas Instruments, Victor, Wang and other microcomputers running PC™-DOS, MS™-DOS and CP/M86™.

Call LIFEBOAT at 212-860-0300 for free information on the Lattice C family of software development tools.

Lifeboat offers Lattice C with a tested set of software tools to provide a complete development system including:

HALO™
PANEL™
PMATE™
PLINK™-86
C-FOOD SMORGASBORD™
LATTICE WINDOW™
FLOAT87™

Color graphic primatives
Screen design aid
Customizable program editor
Overlay linkage editor
Screen and I/O utilities
Multi-window functions
8087 floating point math

LIFEBOAT
Associates

1651 Third Avenue
New York, NY 10028
212-860-0300

Please send me free information on:

- ☐ Lattice and development tools
- ☐ How to get your software published
- ☐ Corporate purchase program
- ☐ Dealer program
- ☐ OEM agreements

☐ Send me the complete LIFEBOAT software catalog. $3.00 enclosed for postage and handling.

Name _____

Company _____

Address _____

City _____

State _____ Zip _____

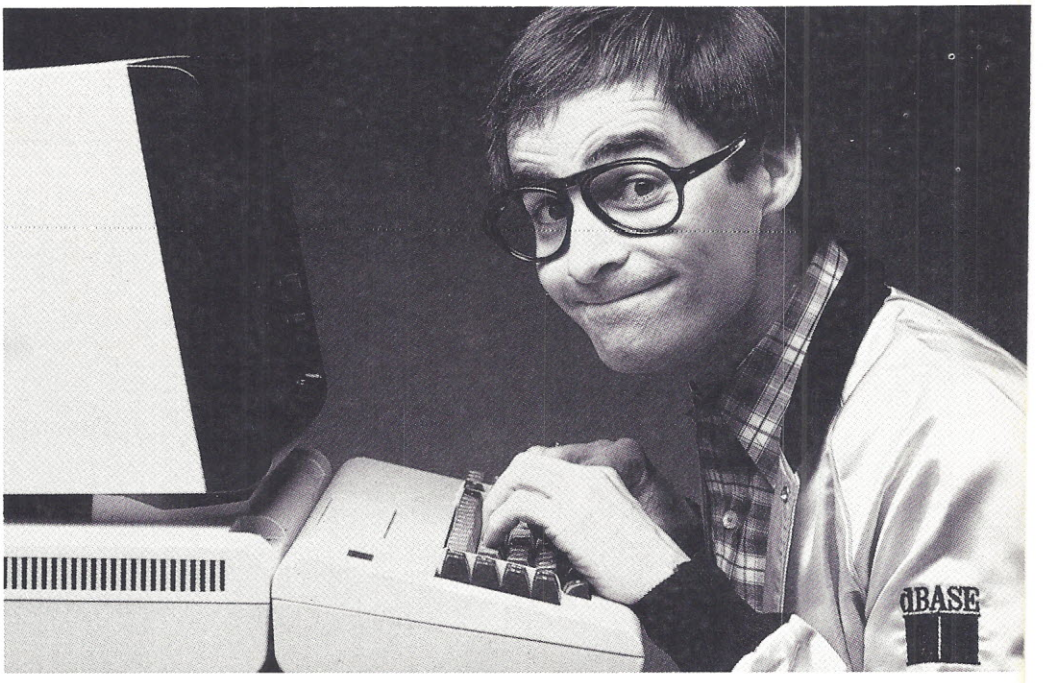
Telephone _____

LATTICE, C-FOOD SMORGASBORD and LATTICE WINDOW, TM Lattice, Inc.

LIFEBOAT, TM Lifeboat Associates
HALO, TM Media Cybernetics
PANEL, TM Roundhill Computer, Ltd
PMATE and PLINK, TM Phoenix Software

FLOAT87, TM Microfloat
IBM and PC, ® TM International Business Machines
MS, TM Microsoft
CP/M86, TM Digital Research

All you dBASE II™ hotshots are about to get what you deserve.



You've written all those slick dBASE II programs.

Business and personal programs. Scientific and educational applications. Packages for just about every conceivable information handling need.

And everybody who sees them loves them because they're so powerful, friendly and easy to use.

But that's just not good enough.

Uh-uh.

Because now you can get the gold and the glory that you really deserve.

Here's how.

We've just released our dBASE II RunTime™ application development module.

And it can turn you into an instant software publisher.

The RunTime module condenses and encodes your source files, protecting your special insights and techniques, so you can sell your code without giving the show away.

RunTime also protects your margins and improves your price position in the marketplace. If your client has dBASE II, all he needs is your encoded application. If not, all you need to install your application is the much less expensive RunTime module.

We'll tell the world.

With your license for the dBASE II RunTime module, we provide labels that identify your program as a dBASE II application, and you get the benefit of all the dBASE II marketing efforts.

We'll also provide additional "how to" information to get you off and running as a software publisher sooner.

And we'll make your products part of our Marketing Referral Service. Besides putting you on our referral hotline, we'll publish your program descriptions and contact information in *dBASE II Applied*, a directory now in computer stores world-wide.

Go for it.

But we can't do any of this until we hear from you.

For details, write RunTime Applications Development, Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230.

Or better yet, just call (213) 204-5570. And get what you deserve today.

dBASE™ II

ASHTON·TATE